Software version TC7.3
OCTOBER 2015

Application Programmer Interface (API) Reference Guide

Cisco TelePresence SX80

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

1

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/sx-docs

Table of Contents

2

Chapter 1

Introduction

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

4

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 SX80

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)
·   Quick reference guides for the TelePresence systems
·   Administrator guides for the TelePresence systems
·   Camera user guide for the PrecisionHD cameras
·   API reference guides
·   TC Console user guide
·   Regulatory compliance and safety information guides
·   Licensing information for the TC software

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

5

What's new in this version

This section provides an overview of the new and changed system settings and new features in the TC7.3 software version.

Software release notes

For a complete overview of new features and changes, we recommend reading the Software Release Notes (TC7). Go to:

▸ http://www.cisco.com/c/en/us/support/collabo-ration-endpoints/telepresence-quick-set-series/products-release-notes-list.html

Software download

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

New features and improvements

Local preview of presentation in a call

This allows the user to preview the presentation locally before sharing it with far end. The feature has previously been available for EX systems, and is now available across the portfolio.

Multiple presentation outside a call

When the system is not in a call, it can simultaneously display multiple external sources on the connected screens, for example from two laptops.

Feature updates

Several feature improvements have been added to better align with C series functionality. There is now support for:

·   H323 / SIP dual registration

·   MultiWay

·   Additional audio call

Snap to Whiteboard feature

It is possible to configure a set up for a whiteboard scenario when using the SpeakerTrack 60.

When the system detects a person speaking close to a whiteboard, the camera will go to a pre-defined preset covering the whiteboard area as defined by the administrator or installer.

There is a setup wizard for the Snap to Whiteboard feature in the administrator settings on the Touch 10.

TC7.3.0-TC7.3.2: Users are notified when snapshots are taken

Both the on screen display and web interface have warnings when the snapshots feature is enabled. A notification pops up on the on screen display, when a snapshot is taken. On the web interface the administrator is warned that this notice will show up when the feature is enabled.

The system also logs when snapshots are taken, and which IP address the request was initiated from.

It is possible to allow and disallow snapshots remotely, but not to observe the room without the users being notified.

TC7.3.3 and later: Remote Monitoring option key

Due to security reasons, taking snapshots of local and far end video streams from the call control page on the system's web interface now requires an option key to be installed on the endpoint.

The remote monitoring option key can only be added to systems that are upgraded to TC7.3.3 and above. Remote monitoring is enabled once the option key is added, and the system rebooted. Once this feature is enabled, the only way to disable it is to remove the option key.

This feature does not display warning messages or indicators on the local system that someone is monitoring the room. Please provide adequate notice to users of the system that the system administrator may monitor and control the camera and screen.

TC7.3.4 and later: Setting an administrator password

The `xCommand SystemUnit AdminPassword Set` command will now change the password of the authenticated user (granted it has administrator rights).

In previous software versions, this command changed the password for the default user account (*admin*).

System configuration changes

New configurations

Cameras Camera [n] AssignedSerialNumber

Cameras Preset TriggerAutofocus

Cameras SpeakerTrack ConnectorDetection Mode

Cameras SpeakerTrack ConnectorDetection CameraLeft

Cameras SpeakerTrack ConnectorDetection CameraRight

Cameras SpeakerTrack Whiteboard Mode

Conference [1..1] DefaultCall Protocol

H323 Profile [1..1] Encryption KeySize

NetworkServices CDP Mode

NetworkServices MultiWay Address

NetworkServices MultiWay Protocol

NetworkServices UPnP Mode *(TC7.3.4)*

NetworkServices UPnP Timeout *(TC7.3.4)*

Configurations that are removed

H323 Profile [1..1] Encryption MinKeySize

Video AllowWebSnapshots *(TC7.3.3)*

Configurations that are modified

Audio Input HDMI [n] VideoAssociation VideoInputSource

OLD: <1/2/3/4>

NEW: <1/2/3/4/5>

Audio Input Line [n] VideoAssociation VideoInputSource

OLD: <1/2/3>

NEW: <1/2/3/4/5>

Audio Input Microphone [n] VideoAssociation VideoInputSource

OLD: <1/2/3>

NEW: <1/2/3/4/5>

Cameras SpeakerTrack TrackingMode *(TC7.3.3)*

OLD: <Default/Fast>

NEW: <Default/Conservative>

Conference [1..1] Multipoint Mode

OLD: <Auto/Off/MultiSite/CUCMMediaResourceGroupList>

NEW: <Auto/Off/MultiSite/MultiWay/CUCMMediaResourceGroupList>

FacilityService Service [1..5] Name

OLD: <S: 0, 255>

NEW: <S: 0, 1024>

FacilityService Service [1..5] Number

OLD: <S: 0, 255>

NEW: <S: 0, 1024>

Phonebook Server [1..1] Type

OLD: <VCS/TMS/Callway/CUCM>

NEW: <VCS/TMS/CUCM>

Provisioning Mode

OLD: <Off/TMS/VCS/CallWay/CUCM/Auto/Edge>

NEW: <Off/TMS/VCS/CUCM/Auto/Edge>

Video AllowWebSnapshots *(removed in TC7.3.3)*

OLD: <Off/On>, default Off

NEW: <Off/On/LocalDeviceOnly>, default LocalDeviceOnly

Video Output Connector [n] MonitorRole *(TC7.3.4)*

OLD: <Auto/First/Second/Third/PresentationOnly>

NEW: <Auto/First/Second/Third/PresentationOnly/Recorder>

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

7

System command changes

New commands

Camera Preset Snapshot Get

Camera Preset Snapshot Remove

Camera Preset Snapshot Store

Cameras SpeakerTrack Whiteboard ActivatePosition

Cameras SpeakerTrack Whiteboard AlignPosition

Cameras SpeakerTrack Whiteboard SetDistance

Cameras SpeakerTrack Whiteboard StorePosition

SystemUnit OptionKey List *(TC7.3.3)*

SystemUnit OptionKey Remove *(TC7.3.3)*

Video Matrix Assign

Video Matrix Reset

Video Matrix Swap

Video Matrix Unassign

Commands that are modified

Audio LocalOutput Add *(TC7.3.3)*
  NEW: VolumeControlled: <On/Off>

Audio LocalOutput Update *(TC7.3.3)*
  NEW: VolumeControlled: <On/Off>

CallHistory Get
  OLD: Filter: <All/Missed/AnsweredElsewhere/Forwarded/ Placed/NoAnswer/Received/UnacknowledgedMissed>
  NEW: Filter: <All/Missed/AnsweredElsewhere/ Forwarded/Placed/NoAnswer/Received/Rejected/ UnacknowledgedMissed>

CallHistory Recents
  OLD: Filter: <All/Missed/AnsweredElsewhere/Forwarded/ Placed/NoAnswer/Received/UnacknowledgedMissed>
  NEW: Filter: <All/Missed/AnsweredElsewhere/ Forwarded/Placed/NoAnswer/Received/Rejected/ UnacknowledgedMissed>

Camera Preset Store
  NEW: TakeSnapshot <False/True>

Presentation Start
  NEW: Instance: <New/1/2/3/4/5/6>
  NEW: OutputId: <1 – 4>

Presentation Stop
  NEW: Instance: <New/1/2/3/4/5/6>

UserInterface ScreenShot Get
  OLD: ScreenShotId: 0 – 35
  NEW: ScreenShotId: 0 – 50

UserInterface ScreenShot Remove
  OLD: ScreenShotId: 0 – 255
  NEW: ScreenShotId: 0 – 50

UserInterface ScreenShot Request
  OLD: ScreenShotId: 0 – 35
  NEW: ScreenShotId: 0 – 50

UserInterface ScreenShot Store
  OLD: ScreenShotId: 0 – 255
  NEW: ScreenShotId: 0 – 50

System status changes

New statuses

Audio Output LocalOutput [n] VolumeControlled *(TC7.3.4)*

Call [n] AttendedTransferFrom

Cameras SpeakerTrack Whiteboard [n] Camera [n] Defined

Cameras SpeakerTrack Whiteboard [n] Camera [n] Pan

Cameras SpeakerTrack Whiteboard [n] Camera [n] Tilt

Cameras SpeakerTrack Whiteboard [n] Camera [n] Zoom

Cameras SpeakerTrack Whiteboard [n] Distance

Conference ActiveSpeaker Manual SiteId

Conference Presentation Instance [n] LocalSendingMode

Conference Presentation Instance [n] LocalSource

MediaChannels Call [n] IncomingAudioChannel [n] Transport RTCP Local Protocol

MediaChannels Call [n] IncomingAudioChannel [n] Transport RTCP Remote Protocol

MediaChannels Call [n] IncomingAudioChannel [n] Transport RTP Local Protocol

MediaChannels Call [n] IncomingAudioChannel [n] Transport RTP Remote Protocol

MediaChannels Call [n] IncomingVideoChannel [n] Transport RTCP Local Protocol

MediaChannels Call [n] IncomingVideoChannel [n] Transport RTCP Remote Protocol

MediaChannels Call [n] IncomingVideoChannel [n] Transport RTP Local Protocol

MediaChannels Call [n] IncomingVideoChannel [n] Transport RTP Remote Protocol

MediaChannels Call [n] OutgoingAudioChannel [n] Transport RTCP Local Protocol

MediaChannels Call [n] OutgoingAudioChannel [n] Transport RTCP Remote Protocol

MediaChannels Call [n] OutgoingAudioChannel [n] Transport RTP Local Protocol

MediaChannels Call [n] OutgoingAudioChannel [n] Transport RTP Remote Protocol

MediaChannels Call [n] OutgoingVideoChannel [n] Transport RTCP Local Protocol

MediaChannels Call [n] OutgoingVideoChannel [n] Transport RTCP Remote Protocol

MediaChannels Call [n] OutgoingVideoChannel [n] Transport RTP Local Protocol

MediaChannels Call [n] OutgoingVideoChannel [n] Transport RTP Remote Protocol

SystemUnit Software OptionKeys RemoteMonitoring *(TC7.3.3)*

Video Layout Prediction Site [n] Family [n] OutputRole [n] Frame [n] ParticipantId

Video Layout Prediction Site [n] Family [n] OutputRole [n] Frame [n] VideoSourceInstance

Video Output HDMI [n] ConnectedDevice CEC [n] DeviceType

Video Output HDMI [n] ConnectedDevice CEC [n] PowerControl

Video Output HDMI [n] ConnectedDevice CEC [n] PowerStatus

Statuses that are modified

SystemUnit Diagnostics Message [n] Type

OLD: <CUCMVendorConfigurationFile/ CallProtocolDualStackConfig/ CallProtocolIPStackPlatformCompatibility/ CallProtocolVcsProvisioningCompatibility/ CameraPairing/CameraSoftwareVersion/CameraStatus/ CamerasDetected/DefaultCallProtocolRegistered/ EthernetDuplexMatches/H320GatewayStatus/ H323GatekeeperStatus/HasValidReleaseKey/ IPv4Assignment/IPv6Assignment/IPv6Mtu/ ISDNLinkCompatibility/ISDNLinkIpStack/ NTPStatus/NetSpeedAutoNegotiated/ OSDVideoOutput/ProvisioningStatus/SIPEncryption/ SIPListenPortAndOutboundMode/SIPProfileRegistration/ SIPProfileType/SelectedVideoInputSourceConnected/ SipIceAndAnatConflict/TLSVerifyRequiredCerts/ TouchPanelConnection/TurnBandwidth/ UdpPortRangeViolation/ValidAdminPassword/ VideoFromInternalCamera/VideoInputStability>

NEW: <CUCMVendorConfigurationFile/ CallProtocolDualStackConfig/ CallProtocolIPStackPlatformCompatibility/ CallProtocolVcsProvisioningCompatibility/ CameraPairing/CameraSoftwareVersion/CameraStatus/ CamerasDetected/DefaultCallProtocolRegistered/ EthernetDuplexMatches/H320GatewayStatus/ H323GatekeeperStatus/HasValidReleaseKey/ IPv4Assignment/IPv6Assignment/IPv6Mtu/ ISDNLinkCompatibility/ISDNLinkIpStack/ NTPStatus/NetSpeedAutoNegotiated/ OSDVideoOutput/ProvisioningStatus/SIPEncryption/ SIPListenPortAndOutboundMode/SIPProfileRegistration/ SIPProfileType/SelectedVideoInputSourceConnected/ SipIceAndAnatConflict/TLSVerifyRequiredCerts/ TouchPanelConnection/TurnBandwidth/ UdpPortRangeViolation/ValidPasswords/ VideoFromInternalCamera/VideoInputStability>

NEW *(TC7.3.4):* <CUCMVendorConfigurationFile/ CallProtocolDualStackConfig/ CallProtocolIPStackPlatformCompatibility/ CallProtocolVcsProvisioningCompatibility/ CameraPairing/CameraSoftwareVersion/CameraStatus/ CamerasDetected/DefaultCallProtocolRegistered/ EthernetDuplexMatches/H320GatewayStatus/ H323GatekeeperStatus/HasValidReleaseKey/ IPv4Assignment/IPv6Assignment/IPv6Mtu/ ISDNLinkCompatibility/ISDNLinkIpStack/ NTPStatus/NetSpeedAutoNegotiated/ OSDVideoOutput/ProvisioningStatus/SIPEncryption/ SIPListenPortAndOutboundMode/SIPProfileRegistration/ SIPProfileType/SelectedVideoInputSourceConnected/ SipIceAndAnatConflict/TLSVerifyRequiredCerts/ TouchPanelConnection/TurnBandwidth/ UdpPortRangeViolation/ValidPasswords/ VideoFromInternalCamera/VideoInputStability/ TemperatureCheck>

9

Chapter 2

About the API

10

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 is 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 config-uration 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 regardless of an IP address, DNS or network configuration.

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

Connect a standard 9-pin serial cable <115200, N, 8, 1> to the COM port on the SX80 codec.

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 115200. The codec can give very much feedback. We therefore recommend keeping the connection at this speed.

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 communi-cating 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 21 in this document.

Ethernet port

You can connect a non-Cisco controller directly to an SX80 codec. Connect your device to the network port 2 or 3 with an Ethernet cable. This way the device gets a local IP address from the SX80, and is not part of the corporate network.

This method can also be used to connect to the codec with SSH. In this case you can use the IP address 169.254.1.1.

Note: Only one non-Cisco device can be connected to the network ports at a time.

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 top level 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        xcommand      xconfiguration      xevent        xfeedback
xgetxml     xdocument     xpreferences        xstatus       bye
echo        log           systemtools
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 16.

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 15.

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 17.

xFeedback

The Feedback commands are used to specify what parts of the configuration and status hierarchies to monitor. Feedback is only issued on the RS-232, 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 19.

xPreferences

The xPreferences command is used to set preferences for the RS-232, Telnet and SSH sessions.
Refer to ▸"API output" on page 12.

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 18.

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 19.

Other commands

Systemtools

The systemtools commands are a set of command line tools for administrative control and diagnostics. The commands can be used for advanced troubleshooting together with Cisco technical support. Systemtools are not a part of the programming API. Refer to ▸"The SystemTools commands" on page 228.

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 words that starts with OUT and HDMI:

```
xconfiguration //out//hdmi
*c xConfiguration Audio Output HDMI 1 Level: 0
*c xConfiguration Audio Output HDMI 1 Mode: On
*c xConfiguration Audio Output HDMI 2 Level: 0
*c xConfiguration Audio Output HDMI 2 Mode: Off
** end

OK
```

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: 1920
*s Video Input Source 2 Resolution Width: 0
*s Video Input Source 3 Resolution Width: 0
*s Video Input Source 4 Resolution Width: 0
*s Video Output Connector 1 Resolution Width: 1920
*s Video Output Connector 2 Resolution Width: 1280
*s Video Output Connector 3 Resolution Width: 1280
** end

OK
```

Example 3: Value types and formats

```
xconfiguration ??
*? xConfiguration Audio DefaultVolume: <0..100>
*? xConfiguration Audio Input HDMI[1] Level: <-24..0>
*? xConfiguration Audio Input HDMI[1] Mode: <Off/On>
*? xConfiguration Audio Input HDMI[1] VideoAssociation MuteOnInactiveVideo: <Off/On>
*? xConfiguration Audio Input HDMI[1] VideoAssociation VideoInputSource: <1, 2, 3, 4, 5>
*? xConfiguration Audio Input HDMI[2] Level: <-24..0>
*? xConfiguration Audio Input HDMI[2] Mode: <Off/On>
*? xConfiguration Audio Input HDMI[2] VideoAssociation MuteOnInactiveVideo: <Off/On>
*? xConfiguration Audio Input HDMI[2] VideoAssociation VideoInputSource: <1, 2, 3, 4, 5>
.
.
.
*? xConfiguration Video SelfviewDefault Mode: <Off/Current/On>
*? xConfiguration Video SelfviewDefault OnMonitorRole: <First/Second/Current/Third>
*? xConfiguration Video SelfviewDefault PIPPosition: <Current/UpperLeft/UpperCenter/
UpperRight/CenterLeft/CenterRight/LowerLeft/LowerRight>
*? xConfiguration Video Wallpaper: <None/Custom> DEPRECATED

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              Conference         Key                Security
Bookings           Dial               Logging            SStringSend
Boot               DTMFSend           Message            Standby
Call               Experimental       Peripherals        SystemUnit
CallHistory        FacilityService    Phonebook          TStringSend
CamCtrlPip         FarEndControl      Presentation       UserInterface
Camera             GPIO               Preset             UserManagement
Cameras            HttpFeedback       Provisioning       Video

OK
```

```
xCommand ??
xCommand Audio Equalizer List
     EqualizerId: <1..8>
xCommand Audio Equalizer Update
     EqualizerId(r): <1..8>
     FilterType(r): <highpass/highshelf/lowpass/lowshelf/none/peaking>
     Frequency(r): <S: 0, 32>
     Gain(r): <S: 0, 32>
     Q(r): <S: 0, 32>
     Section(r): <1..6>
.
.
.
xCommand Video PreviewFilmstrip Set
     Position(r): <LowerCenter/LowerLeft/LowerRight/UpperCenter/UpperLeft/UpperRight>
xCommand Video Selfview Set
     FullscreenMode: <On/Off>
     Mode: <On/Off>
     OnMonitorRole: <First/Fourth/Second/Third>
     PIPPosition: <CenterLeft/CenterRight/LowerLeft/LowerRight/UpperCenter/UpperLeft/UpperRight>

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              Provisioning       SystemUnit

Cameras            Logging           RTP                Time

Conference         Network           Security           UserInterface

Experimental       NetworkServices   SerialPort         Video

FacilityService    Peripherals       SIP

GPIO               Phonebook         Standby


OK
```

```
xConfiguration ??
*? xConfiguration Audio DefaultVolume: <0..100>
*? xConfiguration Audio Input HDMI[1] Level: <-24..0>
*? xConfiguration Audio Input HDMI[1] Mode: <Off/On>
*? xConfiguration Audio Input HDMI[1] VideoAssociation MuteOnInactiveVideo: <Off/On>
*? xConfiguration Audio Input HDMI[1] VideoAssociation VideoInputSource: <1, 2, 3, 4, 5>
*? xConfiguration Audio Input HDMI[2] Level: <-24..0>
*? xConfiguration Audio Input HDMI[2] Mode: <Off/On>
*? xConfiguration Audio Input HDMI[2] VideoAssociation MuteOnInactiveVideo: <Off/On>
.
.
.
*? xConfiguration Video PIP Presentation DefaultValue Position: <Current/UpperLeft/UpperCenter/UpperRight/
 CenterLeft/CenterRight/LowerLeft/LowerRight>
*? xConfiguration Video SelfviewDefault FullscreenMode: <Off/Current/On>
*? xConfiguration Video SelfviewDefault Mode: <Off/Current/On>
*? xConfiguration Video SelfviewDefault OnMonitorRole: <First/Second/Current/Third>
*? xConfiguration Video SelfviewDefault PIPPosition: <Current/UpperLeft/UpperCenter/UpperRight/CenterLeft/
 CenterRight/LowerLeft/LowerRight>
*? xConfiguration Video Wallpaper: <None/Custom> DEPRECATED


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           GPIO            Network          Standby
Call            H320            NetworkServices  SystemUnit
Camera          H323            Peripherals      Time
Cameras         HttpFeedback    Preset           UserInterface
Conference      ICE             Provisioning     Video
Diagnostics     Logging         Security
Experimental    MediaChannels   SIP
OK
```

Addressing status information with xStatus

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

```
xStatus <address expression>
```

Example 1: Query all ongoing Call information:

```
xStatus Call
*s Call 2 Status: Connected
*s Call 2 AnswerState: Answered
*s Call 2 CallPriority: None
*s Call 2 CallType: Video
*s Call 2 CallbackNumber: "sip:name@company.com"
*s Call 2 DeviceType: Endpoint
*s Call 2 Direction: Outgoing
*s Call 2 DisplayName: "Display Name"
*s Call 2 Duration: 15
*s Call 2 Encryption Type: "None"
*s Call 2 FacilityServiceId: 0
*s Call 2 ModifyState: Idle
*s Call 2 PlacedOnHold: False
*s Call 2 Protocol: "sip"
*s Call 2 ReceiveCallRate: 6000
*s Call 2 RemoteNumber: "name@company.com"
*s Call 2 TransmitCallRate: 4000
** end
```

Example 2: Query the protocol for a call:

```
xStatus Call Protocol
*s Call 2 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
```

Example 5: TString message received

TString message received is an event reported when far end has sent a TString message.

```
*e TString CallId: 132 Message: "ee"

** end
```

Example 6: SString message received

SString message received is an event reported when far end has sent a SString message.

```
*e SString String: "ee" Id: 132

** 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. See ▸"Feedback from codec over HTTP" on page 22 for more information.

> 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: Name of all video input connectors.

Terminal query

```
xConfiguration Video Input Connector Name
*c xConfiguration Video Input Connector 1 Name:
"NameA"
*c xConfiguration Video Input Connector 2 Name:
"NameB"
*c xConfiguration Video Input Connector 3 Name:
"NameC"
*c xConfiguration Video Input Connector 4 Name:
"NameD"
*c xConfiguration Video Input Connector 5 Name:
"NameE"
** end
```

Equivalent feedback expression

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

Example 3: Name of video input connector 3.

Terminal query

```
xConfiguration Video Input Connector 3 Name
*c xConfiguration Video Input Connector 3 Name:
"NameC"
** end
```

Equivalent feedback expression

```
xFeedback register /Configuration/Video/Input/
Connector[@item='3']/Name
```

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><DefaultVolume>80</
DefaultVolume></Audio><UserInterface><Wallpaper>C
ustom</Wallpaper></UserInterface></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>
```

    21

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, 2048>

    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. See the ▶"Feedback mechanism" on page 19 section for more information about the expression formats.

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 the input source type for video input connector 2.

Terminal

```
xConfiguration Video Input Connector 2
InputSourceType: camera
```

XML

```
<Configuration>

  <Video>

    <Input>

      <Connector item="2">

        <InputSourceType>camera</InputSourceType>

      </Connector>

    </Input>

  </Video>

</Configuration>
```

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.　　　　23

Dos and don'ts

Here are some best practices 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 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

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

25

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/sx-docs

Audio configuration

xConfiguration Audio Input HDMI [1..3] Mode

Determine if the audio channels on the HDMI input shall be enabled. The HDMI input has two audio channels.

As default, Audio is disabled on HDMI [1..2] and enabled on HDMI [3].

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 Input HDMI [1..3] Level

Define the audio level of the HDMI input connector, in steps of 1 dB.

Requires user role: ADMIN

Value space: <-24..0>

Range: Select a value between -24 and 0, in steps of 1 dB.

Example: `xConfiguration Audio Input HDMI 2 Level: 0`

xConfiguration Audio Input HDMI [1..3] VideoAssociation VideoInputSource

It is possible to associate an audio source with a video source, and further to determine whether to play or mute audio depending on whether the video source is presented or not. By default, audio is not muted.

Use the Audio Input HDMI [n] VideoAssociation VideoInputSource setting to define which video source to associate the audio source with. Use the Audio Input HDMI [n] VideoAssociation MuteOnInactiveVideo setting to define whether to play or mute audio when not presenting the video source.

Requires user role: ADMIN

Value space: <1/2/3/4/5>

Range: Select one of the video input sources.

Example: `xConfiguration Audio Input HDMI 3 VideoAssociation VideoInputSource: 1`

xConfiguration Audio Input HDMI [1..3] VideoAssociation MuteOnInactiveVideo

It is possible to associate an audio source with a video source, and further to determine whether to play or mute audio depending on whether the video source is presented or not. By default, audio is not muted.

Use the Audio Input HDMI [n] VideoAssociation VideoInputSource setting to define which video source to associate the audio source with. Use the Audio Input HDMI [n] VideoAssociation MuteOnInactiveVideo setting to define whether to play or mute audio when not presenting the video source.

Requires user role: ADMIN

Value space: <Off/On>

Off: The audio source is not associated with a video source. The audio will be played locally and to far end regardless of whether the video source is presented.

On: The audio source is associated with a video source. The audio will be played (locally and to far end) when the associated video source is presented. The audio will be muted when the video source is not presented.

Example: `xConfiguration Audio Input HDMI 3 VideoAssociation MuteOnInactiveVideo: Off`

xConfiguration Audio Input Line [1..4] Equalizer ID

Select the audio input line equalizer ID.

Requires user role: ADMIN

Value space: <1..8>

Range: Select EqualizerID 1 to 8.

Example: `xConfiguration Audio Input Line 1 Equalizer ID: 1`

xConfiguration Audio Input Line [1..4] Equalizer Mode

Set the audio input line equalizer mode.

Requires user role: ADMIN

Value space: <Off/On>

Off: No equalizer.

On: Enable the equalizer for the audio input line.

Example: `xConfiguration Audio Input Line 1 Equalizer Mode: Off`

xConfiguration Audio Input Line [1..4] VideoAssociation VideoInputSource

It is possible to associate an audio source with a video source, and further to determine whether to play or mute audio depending on whether the video source is presented or not. By default, audio is not muted.

Use the Audio Input Line [n] VideoAssociation VideoInputSource setting to define which video source to associate the audio source with. Use the Audio Input Line [n] VideoAssociation MuteOnInactiveVideo setting to define whether to play or mute audio when not presenting the video source.

Requires user role: ADMIN

Value space: <1/2/3/4/5>

Range: Select one of the video input sources.

Example: `xConfiguration Audio Input Line 1 VideoAssociation VideoInputSource: 1`

xConfiguration Audio Input Line [1..4] VideoAssociation MuteOnInactiveVideo

It is possible to associate an audio source with a video source, and further to determine whether to play or mute audio depending on whether the video source is presented or not. By default, audio is not muted.

Use the Audio Input Line [n] VideoAssociation VideoInputSource setting to define which video source to associate the audio source with. Use the Audio Input Line [n] VideoAssociation MuteOnInactiveVideo setting to define whether to play or mute audio when not presenting the video source.

Requires user role: ADMIN

Value space: <Off/On>

Off: The audio source is not associated with a video source. The audio will be played locally and to far end regardless of whether the video source is presented.

On: The audio source is associated with a video source. The audio will be played (locally and to far end) when the associated video source is presented. The audio will be muted when the video source is not presented.

Example: `xConfiguration Audio Input Line 1 VideoAssociation MuteOnInactiveVideo: Off`

xConfiguration Audio Input Line [1..4] Channel

Define whether the Audio Line input is a mono signal or part of a multichannel signal.

Requires user role: ADMIN

Value space: <Right/Left/Mono>

Right: The Audio Line input signal is the right channel of a stereo signal.

Left: The Audio Line input signal is the left channel of a stereo signal.

Mono: The Audio Line input signal is a mono signal.

Example: `xConfiguration Audio Input 1 Channel: Left`

xConfiguration Audio Input Line [1..4] Level

Define the level of the audio source on the line input connector.

See the Audio Level table in the Physical Interface Guide for the codec for a complete overview of the values represented in dB.

Requires user role: ADMIN

Value space: <0..24>

Range: Select a value between 0 to 24, in steps of 1 dB.

Example: `xConfiguration Audio Input Line 1 Level: 10`

xConfiguration Audio Input Line [1..4] Mode

Set the audio input line mode.

Requires user role: ADMIN

Value space: <Off/On>

Off: Disable the Audio Line input.

On: Enable the Audio Line input.

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

xConfiguration Audio Input Microphone [1..8] EchoControl Mode

The echo canceller continuously adjusts itself to the audio characteristics of the room and compensate for any changes it detects in the audio environment. If the changes in the audio conditions are very significant the echo canceller may take a second or two to re-adjust.

Requires user role: ADMIN

Value space: <Off/On>

Off: Echo Control should be switched Off if external echo cancellation or playback equipment is used.

On: Echo Control is normally set to On to prevent the far end from hearing their own audio. Once selected, echo cancellation is active at all times.

Example: `xConfiguration Audio Input Microphone 1 EchoControl Mode: On`

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

32

xConfiguration Audio Input Microphone [1..8] EchoControl NoiseReduction

The system has a built-in noise reduction which reduces constant background noise (for example noise from air-conditioning systems, cooling fans etc.). In addition, a high pass filter (Humfilter) reduces very low frequency noise. Requires the Echo Control Mode to be enabled for the microphone.

Requires user role: ADMIN

Value space: <Off/On>

Off: Turn off the Noise Reduction.

On: The Noise Reduction should be enabled in the presence of low frequency noise.

Example: `xConfiguration Audio Input Microphone 1 EchoControl NoiseReduction: On`

xConfiguration Audio Input Microphone [1..8] EchoControl Dereverberation

The system has built-in signal processing to reduce the effect of room reverberation. Requires the Echo Control Mode to be enabled for the microphone.

Requires user role: ADMIN

Value space: <Off/On>

Off: Turn off the dereverberation.

On: Turn on the dereverberation.

Example: `xConfiguration Audio Input Microphone 1 EchoControl Dereverberation: On`

xConfiguration Audio Input Microphone [1..8] Equalizer ID

Select the audio input microphone equalizer ID.

Requires user role: ADMIN

Value space: <1..8>

Range: Select Equalizer ID 1 to 8.

Example: `xConfiguration Audio Input Microphone 1 Equalizer ID: 1`

xConfiguration Audio Input Microphone [1..8] Equalizer Mode

Set the audio input microphone equalizer mode.

Requires user role: ADMIN

Value space: <Off/On>

Off: No equalizer.

On: Enable the equalizer for the audio input microphone.

Example: `xConfiguration Audio Input Microphone 1 Equalizer Mode: Off`

xConfiguration Audio Input Microphone [1..8] VideoAssociation MuteOnInactiveVideo

Enable association of a video source to a microphone audio input.

Requires user role: ADMIN

Value space: <Off/On>

Off: No video source is associated.

On: A video source is associated, and the audio will be muted if the associated video source is not displayed.

Example: `xConfiguration Audio Input Microphone 1 VideoAssociation MuteOnInactiveVideo: On`

xConfiguration Audio Input Microphone [1..8] VideoAssociation VideoInputSource

Select the associated video input source.

Requires user role: ADMIN

Value space: <1/2/3/4/5>

Range: Select one of the video input sources.

Example: `xConfiguration Audio Input Microphone 1 VideoAssociation VideoInputSource: 1`

xConfiguration Audio Input Microphone [1..8] Level

Define the audio level of the Microphone input connector.

See the Audio Level table in the Physical Interface Guide for the codec for a complete overview of the values represented in dB.

Requires user role: ADMIN

Value space: <0..70>

Range: Select a value between 0 and 70, in steps of 1 dB.

Example: `xConfiguration Audio Input Microphone 1 Level: 58`

xConfiguration Audio Input Microphone [1..8] Mode

Set the audio input microphone mode.

Requires user role: ADMIN

Value space: <Off/On>

Off: Disable the microphone connector.

On: Enable the microphone connector.

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

xConfiguration Audio Input Microphone [1..8] Type

The microphone connectors are intended for electret type microphones. The microphone connector can be set to line or microphone mode.

Requires user role: ADMIN

Value space: <Microphone/Line>

Microphone: Select Microphone when you have 48 V Phantom voltage and the pre-amplification is On.

Line: Select Line when you have a standard balanced line input. The phantom voltage and pre-amplification is Off.

Example: `xConfiguration Audio Input Microphone 1 Type: Line`

xConfiguration Audio Output HDMI [1..2] Level

Define the output level of the HDMI output connector, in steps of 1 dB.

Requires user role: ADMIN

Value space: <-24..0>

Range: Select a value between -24 and 0, in steps of 1 dB.

Example: `xConfiguration Audio Output HDMI 1 Level: 0`

xConfiguration Audio Output HDMI [1..2] Mode

Determine if the audio channel on the HDMI output connector shall be enabled.

Requires user role: ADMIN

Value space: <Off/On>

Off: Disable the audio channel on the HDMI output.

On: Enable the audio channel on the HDMI output.

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

xConfiguration Audio Output Line [1..6] Channel

Define whether the Audio Line output is a mono signal or part of a multichannel signal.

Requires user role: ADMIN

Value space: <Right/Left/Mono>

Right: The Audio Line output signal is the right channel of a stereo signal.

Left: The Audio Line output signal is the left channel of a stereo signal.

Mono: The Audio Line output signal is a mono signal.

Example: `xConfiguration Audio Output Line 1 Channel: left`

xConfiguration Audio Output Line [1..6] Equalizer ID

Select the audio output line equalizer ID.

Requires user role: ADMIN

Value space: <1..8>

Range: Select EqualizerID 1 to 8.

Example: `xConfiguration Audio Output Line 1 Equalizer ID: 1`

xConfiguration Audio Output Line [1..6] Equalizer Mode

Set the audio output line equalizer mode.

Requires user role: ADMIN

Value space: <Off/On>

Off: No equalizer.

On: Enable the equalizer for the audio output line.

Example: `xConfiguration Audio Output Line 1 Equalizer Mode: Off`

xConfiguration Audio Output Line [1..6] Level

Define the audio level on the line output connector.

See the Audio Level table in the Physical Interface Guide for the codec for a complete overview of the menu values represented in dB.

Requires user role: ADMIN

Value space: <-24..0>

Range: Select a value between -24 and 0, in steps of 1 dB.

Example: `xConfiguration Audio Output Line 1 Level: -10`

xConfiguration Audio Output Line [1..6] Mode

Set the audio output line mode.

Requires user role: ADMIN

Value space: <Off/On>

Off: Disable the Audio Line output.

On: Enable the Audio Line output.

Example: `xConfiguration Audio Output Line 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 typing text or numbers on the Touch controller.

Requires user role: USER

Value space: <Off/On>

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

On: You will hear key tones when you type.

Example: `xConfiguration Audio SoundsAndAlerts KeyTones Mode: Off`

xConfiguration Audio SoundsAndAlerts RingTone

This setting defines which ringtone to use for incoming calls. You need to enter the exact name of the ringtone. You can find the available ringtones the following ways.

Web interface: On the Configuration > Personalization page.

Touch controller: On the Ringtone & Sound panel of the Settings menu. This panel is either in the open part of the Settings menu, or included in the password protected Administrator menu. The UserInterface UserPreference setting defines which panels will be in the password protected area.

Requires user role: USER

Value space: <S: 1, 100>

Format: String with a maximum of 100 characters.

Example: `xConfiguration Audio SoundsAndAlerts RingTone: "Sunrise"`

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 DefaultVolume

Set the default speaker volume. The volume returns to this value when you switch on or restart the video system. You can also run the following API command to return to the default value: xCommand Audio Volume SetToDefault. Run the xCommand Audio Volume commands or use the Touch controller to change the volume while the video system is running.

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 DefaultVolume: 50`

Cameras configuration

xConfiguration Cameras PowerLine Frequency

If your camera supports power line frequency anti-flickering, the camera is able to compensate for any flicker noise from the electrical power supply. You should set this camera configuration based on your power line frequency. If your camera supports auto detection of line frequency, you can select the Auto option in the configuration.

All Cisco Precision cameras support both anti-flickering and auto detection of line frequency. Auto is the default value, so you should change this setting if you have a camera that does not support auto detection.

Requires user role: ADMIN

Value space: <Auto/50Hz/60Hz>

Auto: Allow the camera to detect the power frequency automatically.

50Hz: Use this value when the power line frequency is 50 Hz.

60Hz: Use this value when the power line frequency is 60 Hz.

Example: `xConfiguration Cameras PowerLine Frequency: Auto`

xConfiguration Cameras Preset TriggerAutofocus

The current position (pan and tilt), zoom and focus are stored with a preset. Use this setting to determine if the camera should refocus or use the focus value that is stored with the preset.

Requires user role: ADMIN

Value space: <Auto/Off/On>

Auto: Whether the camera refocuses or not when selecting a preset, depends on the camera type.

Off: The focus value that is stored with the preset will be used. The camera will not refocus when selecting a preset.

On: The camera will refocus when selecting a preset. The focus value that is stored with the preset may be overridden.

Example: `xConfiguration Cameras Preset TriggerAutofocus: Auto`

xConfiguration Cameras SpeakerTrack Mode

This setting applies only when using a Cisco TelePresence SpeakerTrack 60 camera.

The SpeakerTrack 60 camera assembly consists of two cameras and uses an audio tracking technique that finds and captures a close-up of the active speaker. When a change of speaker is detected, the system can switch automatically between the two cameras to always show the best camera view. Refer to the Cameras SpeakerTrack TrackingMode setting for different switching modes.

Requires user role: USER

Value space: <Auto/Off>

Auto: Speaker tracking is switched on. The cameras in the camera assembly behave as one integrated unit that finds the active speaker and dynamically chooses the best camera view.

Off: The cameras operate as two individual cameras. Speaker tracking is not used.

Example: `xConfiguration Cameras SpeakerTrack Mode: Auto`

xConfiguration Cameras SpeakerTrack TrackingMode

This setting applies only when using a Cisco TelePresence SpeakerTrack 60 camera, and when Cameras SpeakerTrack Mode is set to Auto.

The speaker tracking algorithm can react to changes in two modes, one faster than the other. The mode determines when the camera view will change to a new speaker. In software versions TC7.3.0 and TC7.3.1 the Default value is the most conservative mode; the fastest mode is Default in TC7.3.2 and later.

Requires user role: USER

Value space: TC7.3.0 and TC7.3.1: <Default/Fast>; TC7.3.2 and later: <Conservative/Default>

Default: Normal tracking mode.

Conservative: The camera view will change to a new speaker later than in Normal mode.

Fast: The camera view will change to a new speaker faster than in Normal mode.

Example: `xConfiguration Cameras SpeakerTrack TrackingMode: Default`

xConfiguration Cameras SpeakerTrack ConnectorDetection Mode

This setting applies only when a Cisco TelePresence SpeakerTrack 60 camera is connected to the codec (video system).

Determine whether to automatically detect or manually configure which video input each individual camera is connected to. You should choose manual configuration in situations where the codec does not receive EDID information from the cameras. Typically, this will be when you use HDMI repeaters that do not pass on EDID information.

Requires user role: USER

Value space: <Auto/Manual>

Auto: Automatically detect which video inputs the cameras are connected to.

Manual: Manually define which video inputs the cameras are connected to. Use the Cameras SpeakerTrack ConnectorDetection CameraLeft and Cameras SpeakerTrack ConnectorDetection CameraRight settings.

Example: `xConfiguration Cameras SpeakerTrack ConnectorDetection Mode: Auto`

xConfiguration Cameras SpeakerTrack ConnectorDetection CameraLeft

This setting applies only when a Cisco TelePresence SpeakerTrack 60 camera is connected to the codec (video system). Furthermore, Cameras SpeakerTrack ConnectorDetection Mode must be set to Manual.

Enter the number of the video input that SpeakerTrack 60's left camera is connected to. For example, set to 1 if the left camera is connected to video input 1.

Requires user role: USER

Value space: <1..5>

Format: Select a valid video input number.

Example: `xConfiguration Cameras SpeakerTrack ConnectorDetection CameraLeft: 1`

xConfiguration Cameras SpeakerTrack ConnectorDetection CameraRight

This setting applies only when a Cisco TelePresence SpeakerTrack 60 camera is connected to the codec (video system). Furthermore, Cameras SpeakerTrack ConnectorDetection Mode must be set to Manual.

Enter the number of the video input that SpeakerTrack 60's right camera is connected to. For example, set to 2 if the right camera is connected to video input 2.

Requires user role: USER

Value space: <1..5>

Format: Select a valid video input number.

Example: `xConfiguration Cameras SpeakerTrack ConnectorDetection CameraRight: 2`

xConfiguration Cameras SpeakerTrack Whiteboard Mode

This setting applies only when a Cisco TelePresence SpeakerTrack 60 camera is connected to the codec.

Determine whether to enable the Snap to Whiteboard feature or not.

The Snap to Whiteboard feature relies on a speaker track camera. When a presenter is standing next to the whiteboard, the camera will capture both the presenter and the whiteboard if the Snap to Whiteboard feature is enabled. If the feature is disabled, only the presenter will be captured.

The Snap to Whiteboard feature is set up from the Touch controller.

Requires user role: ADMIN

Value space: <Off/On>

Off: The Snap to Whiteboard feature is disabled.

On: The Snap to Whiteboard feature is enabled.

Example: `xConfiguration Cameras SpeakerTrack Whiteboard Mode: Off`

xConfiguration Cameras Camera [1..7] AssignedSerialNumber

The camera ID is the number n in Camera [n]. By default, the camera ID is assigned automatically to a camera. If EDID information is not passed on from the camera to the codec, the camera ID is not persistent after a reboot. This means that a camera may get a new camera ID when the codec (video system) is restarted.

You should use the Cameras Camera AssignedSerialNumber setting to cater for configurations where the codec does not receive EDID information from multiple cameras. This setting allows you to manually assign a camera ID to a camera by associating the camera ID with the camera's serial number. The setting is persistent until the codec is factory reset.

Typical situations where the codec does not receive EDID information are: when you connect a Cisco TelePresence 60 camera using 3G-SDI; when you connect a Cisco TelePresence 40 (Cisco PrecisionHD 1080p4xS2) camera; when you use an HDMI repeater that does not pass on EDID information.

The default value is an empty string.

Requires user role: USER

Value space: <S: 0, 20>

Format: The camera's serial number. String with a maximum of 20 characters.

Example: `xConfiguration Cameras Camera 1 AssignedSerialNumber: "FTT0123456F"`

xConfiguration Cameras Camera [1..7] 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..7] 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..7] Brightness Level

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

Requires user role: ADMIN

Value space: <1..31>

Range: Select a value between 1 and 31.

Example: xConfiguration Cameras Camera 1 Brightness Level: 20

xConfiguration Cameras Camera [1..7] Flip

With Flip mode (vertical flip) you can flip the image upside down. Flipping applies both to the self-view and the video that is transmitted to the far end.

Requires user role: USER

Value space: <Auto/Off/On>

Auto: If the camera detects that it is mounted upside down, the image is automatically flipped. If the camera cannot auto-detect whether it is mounted upside down or not, the image is not changed.

Off: Display the image on screen the normal way.

On: Display the image flipped upside down. 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: Auto

xConfiguration Cameras Camera [1..7] 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..7] Gamma Mode

This setting enables gamma corrections, and applies only to cameras which support gamma mode. Gamma describes the nonlinear relationship between image pixels and monitor brightness.

Requires user role: ADMIN

Value space: <Auto/Manual>

Auto: Auto is the default and the recommended setting.

Manual: In manual mode the gamma value is changed with the gamma level setting, ref: Cameras Camera [1..n] Gamma Level.

Example: xConfiguration Cameras Camera 1 Gamma Mode: Auto

xConfiguration Cameras Camera [1..7] 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. Requires the Gamma Mode to be set to Manual.

Requires user role: ADMIN

Value space: <0..7>
Range: Select a value between 0 and 7.

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

xConfiguration Cameras Camera [1..7] IrSensor

Not applicable in this version.

xConfiguration Cameras Camera [1..7] Mirror

With Mirror mode (horizontal flip) you can mirror the image on screen. Mirroring applies both to the self-view and the video that is transmitted to the far end.

Requires user role: ADMIN

Value space: <Auto/Off/On>
Auto: If the camera detects that it is mounted upside down, the image is automatically mirrored. If the camera cannot auto-detect whether it is mounted upside down or not, the image is not changed.
Off: Display the image as other people see you.
On: Display the image as you see yourself in a mirror.

Example: `xConfiguration Cameras Camera 1 Mirror: Auto`

xConfiguration Cameras Camera [1..7] 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..7] Whitebalance Mode

Set the camera white balance mode.

Requires user role: ADMIN

Value space: <Auto/Manual>
Auto: The camera will continuously adjust the white balance depending on the camera view.
Manual: Enables manual control of the camera white balance. The white balance level is set using the Cameras Camera Whitebalance Level setting.

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

xConfiguration Cameras Camera [1..7] Whitebalance Level

Set the white balance level. Requires the Camera Whitebalance Mode to be set to manual.

Requires user role: ADMIN

Value space: <1..16>
Range: Select a value between 1 and 16.

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

xConfiguration Cameras Camera [1..7] DHCP

Not applicable for this product.

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

39

Copyright © 2015 Cisco Systems, Inc. All rights reserved.

Conference configuration

xConfiguration Conference [1..1] ActiveControl Mode

Active control is a feature that allows conference participants to administer a conference on Cisco TelePresence Server using the video system's interfaces. Each user can see the partici-pant list, change video layout, disconnect participants, etc. from the interface. The active control feature is enabled by default, provided that it is supported by the infrastructure (Cisco Unified Communications Manager (CUCM) version 9.1.2 or newer, Cisco TelePresence Video Communication Server (VCS) version X8.1 or newer). Change this setting if you want to disable the active control features.

Requires user role: ADMIN

Value space: <Auto/Off>

Auto: Active control is enabled when supported by the infrastructure.

Off: Active control is disabled.

Example: `xConfiguration Conference ActiveControl Mode: Auto`

xConfiguration Conference [1..1] CallProtocolIPStack

Select if the system should enable IPv4, IPv6, or dual IP stack on the call protocol (SIP, H323).

Requires user role: ADMIN

Value space: <Dual/IPv4/IPv6>

Dual: Enables both IPv4 and IPv6 for the call protocol.

IPv4: When set to IPv4, the call protocol will use IPv4.

IPv6: When set to IPv6, the call protocol will use IPv6.

Example: `xConfiguration Conference 1 CallProtocolIPStack: Dual`

xConfiguration Conference [1..1] AutoAnswer Mode

Set the auto answer mode. Use the Conference AutoAnswer Delay setting if you want the system to wait a number of seconds before answering the call, and use the Conference AutoAnswer Mute setting if you want your microphone to be muted when the call is answered.

Requires user role: ADMIN

Value space: <Off/On>

Off: You must answer incomming calls manually by tapping Answer on the Touch controller.

On: The system automatically answers incoming calls, except if you are already in a call. You must always answer or decline incoming calls manually when you are already engaged in a call.

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. 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. Requires that AutoAnswer Mode is switched on.

Requires user role: ADMIN

Value space: <0..50>

Range: Select a value between 0 and 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 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 (Touch controller). 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

Define 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.
NOTE: If the Encryption Option Key is not installed on the video system, the encryption mode is always Off.

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: <Auto/H323/Sip/H320>

Auto: Enables auto-selection of the call protocol based on which protocols are available. If multiple protocols are available, the order of priority is: 1) SIP; 2) H323; 3) H320. If the system cannot register, or the call protocol is not enabled, the auto-selection chooses H323.

H323: All calls are set up as H.323 calls.

Sip: All calls are set up as SIP calls.

H320: All 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: Auto`

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: 1920`

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

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: 10000

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: 10000

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] 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 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 built-in MultiSite allows up to five participants (yourself included) plus one additional audio call.

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

Requires user role: ADMIN

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

Auto: The multipoint method available will be choosen automatically; if none are available the Multipoint Mode will automatically be set to Off. If both MultiWay and MultiSite are available, the MultiWay service takes priority over the built-in MultiSite.

Off: Multiparty conferences are not allowed.

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

MultiWay: Multiparty conferences are set up using the MultiWay service. If MultiWay is chosen when the MultiWay service is not available, the Multipoint Mode will automatically be set to Off. This may occur when the NetworkServices MultiWay Address setting is empty or not properly set.

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 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; the other options are available for system integrators using the API (Application Programming Interface) command set.

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

Enter the name of the 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. The name will show on the facility service call button, which appears when you tap the question mark icon in the top bar. The other services are available for system integrators using the API (Application Programming Interface) command set.

Requires user role: ADMIN

Value space: <S: 0, 1024>

Format: String with a maximum of 1024 characters.

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

xConfiguration FacilityService Service [1..5] Number

Enter the number (URI or phone number) of the 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; the other options are available for system integrators using the API (Application Programming Interface) command set.

Requires user role: ADMIN

Value space: <S: 0, 1024>

Format: String with a maximum of 1024 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; the other options are available for system integrators using the API (Application Programming Interface) command set.

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`

GPIO configuration

H323 configuration

xConfiguration GPIO Pin [1..4] Mode

The four GPIO pins are configured individually. The state can be retrieved by "xStatus GPIO Pin [1..4] State". The default pin state is High (+12 V). When activated as output, they are set to 0 V. To activate them as input, they must be pulled down to 0 V.

Requires user role: ADMIN

Value space: <InputNoAction/OutputManualState/OutputInCall/OutputMicrophonesMuted/ OutputPresentationOn/OutputAllCallsEncrypted/OutputStandbyActive/InputMuteMicrophones>

InputNoAction: The pin state can be set, but no operation is performed.

OutputManualState: The pin state can be set by "xCommand GPIO ManualState Set PinX: <High/ Low>" (to +12 V or 0 V, respectively).

OutputInCall: The pin is activated when in call, deactivated when not in call.

OutputMicrophonesMuted: The pin is activated when microphones are muted, deactivated when not muted.

OutputPresentationOn: The pin is activated when presentation is active, deactivated when presentation is not active.

OutputAllCallsEncrypted: The pin is activated when all calls are encrypted, deactivated when one or more calls are not encrypted.

OutputStandbyActive: The pin is activated when the system is in standby mode, deactivated when no longer in standby.

InputMuteMicrophones: When the pin is activated (0 V), the microphones will be muted. When deactivated (+12 V), the microphones are unmuted.

Example: xConfiguration GPIO Pin 1 Mode: InputNoAction

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 authentication 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 authenti-cate 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. 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. 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. 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] Encryption KeySize

Define the minimum or maximum key size for the Diffie-Hellman key exchange method, which is used when establishing the Advanced Encryption Standard (AES) encryption key.

Requires user role: ADMIN

Value space: <Min1024bit/Max1024bit/Min2048bit>

Min1024bit: The minimum size is 1024 bit.

Max1024bit: The maximum size is 1024 bit.

Min2048bit: The minimum size is 2048 bit.

Example: xConfiguration H323 Profile 1 Encryption MinKeySize: Max1024bit

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. 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 imple-mented 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 signaling.

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. 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

Logging configuration

xConfiguration Logging Mode

Not applicable in this version.

Network configuration

xConfiguration Network [1..1] IPStack

Select if the system should use IPv4, IPv6, or dual IP stack, on the network interface. NOTE: After changing this setting you may have to wait up to 30 seconds before it takes effect.

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: Dual`

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: "2001:0DB8:0000:0000:0000:0000:0002"

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: "2001:0DB8:0000:0000:0000:0000:0001"

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.

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

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

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

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

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

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

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 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 between 576 and 1500 bytes.

Example: xConfiguration Network 1 MTU: 1500

xConfiguration Network [1..1] Speed

Set the Ethernet link speed.

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

Define which IP addresses (IPv4/IPv6) are allowed for remote access to the codec from SSH/Telnet/HTTP/HTTPS. Multiple IP addresses are separated by a white space.

A network mask (IP range) is specified by <ip address>/N, where N is 1-32 for IPv4, and N is 1-128 for IPv6. The /N is a common indication of a network mask where the first N bits are set. Thus 192.168.0.0/24 would match any address starting with 192.168.0, since these are the first 24 bits in the address.

Requires user role: ADMIN

Value space: <S: 0, 255>

Format: String with a maximum of 255 characters.

Example: `xConfiguration Network 1 RemoteAccess Allow: "10.11.2.3 192.168.0.0/24 2001:0db8:0000:0000:0000:ff00:0042:8329 2001:db8:abcd:0012::0/64"`

xConfiguration Network [1..1] VLAN Voice Mode

Set the VLAN voice mode. The VLAN Voice Mode will be set to Auto automatically if you have Cisco UCM (Cisco Unified Communications Manager) as provisioning infrastructure. Note that Auto mode will NOT work if the NetworkServices CDP Mode setting is Off.

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: Auto`

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 between 1 and 4094.

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

NetworkServices configuration

xConfiguration NetworkServices CDP Mode

Enable or disable the CDP (Cisco Discovery Protocol) daemon. Enabling CDP will make the endpoint report certain statistics and device identifiers to a CDP-enabled switch. If CDP is disabled, the Network VLAN Voice Mode: Auto setting will not work.

Requires user role: ADMIN

Value space: <Off/On>

Off: The CDP daemon is disabled.

On: The CDP daemon is enabled.

Example: `xConfiguration NetworkServices CDP Mode: On`

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.

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 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 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 WelcomeText

Choose which information the user should see when logging on to the codec through Telnet/SSH.

Requires user role: ADMIN

Value space: <Off/On>

Off: The welcome text is: Login successful

On: The welcome text is: Welcome to <system name>; Software version; Software release date; Login successful.

Example: `xConfiguration NetworkServices WelcomeText: On`

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 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.

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`

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

56

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 Medianet Metadata

Switch On or Off the capability to tag media flows with metadata related to the Cisco Medianet deployment.

Requires user role: ADMIN

Value space: <Off/On>

Off: Media flows will not be tagged with such metadata.

On: Media flows will be tagged with such metadata.

Example: `xConfiguration NetworkServices Medianet Metadata: Off`

xConfiguration NetworkServices NTP Mode

The Network Time Protocol (NTP) is used to synchronize the system's time and date to a reference time server. The time server will be queried regularly for time updates.

Requires user role: ADMIN

Value space: <Auto/Manual/Off>

Auto: The system will use an NTP server for time reference. As default, the server address will be obtained from the network's DHCP server. If a DHCP server is not used, or if the DHCP server does not provide an NTP server address, the NTP server address that is specified in the NetworkServices NTP Address setting will be used.

Manual: The system will use the NTP server that is specified in the NetworkServices NTP Address setting for time reference.

Off: The system will not use an NTP server. The Network Services NTP Address setting will be ignored.

Example: `xConfiguration NetworkServices NTP Mode: Auto`

xConfiguration NetworkServices NTP Address

The address of the NTP server that will be used when NetworkServices NTP Mode is set to Manual, and when NetworkServices NTP Mode is 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: "0.tandberg.pool.ntp.org"`

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: ReadOnly`

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 CTMS Mode

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

Video systems 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 CTMS 1.9.1. Encryption is addressed in the NetworkServices CTMS Encryption setting.

Requires user role: ADMIN

Value space: <Off/On>

Off: Multiparty conferencing via CTMS is prohibited.

On: Multiparty 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`

xConfiguration NetworkServices UPnP Mode

Fully disable UPnP (Universal Plug and Play), or enable UPnP for a short time period after the video system has been switched on or restarted.

The default operation is that UPnP is enabled when you switch on or restart the video system. Then UPnP is automatically disabled after the timeout period that is defined in the NetworkServices UPnP Timeout setting.

When UPnP is enabled, the video system advertises its presence on the network. The advertisement permits a Touch controller to discover video systems automatically, and you do not need to manually enter the video system's IP address in order to pair the Touch controller.

Requires user role: ADMIN

Value space: <Off/On>

Off: UPnP is disabled. The video system does not advertise its presence, and you have to enter the video system's IP address manually in order to pair a Touch controller to the video system.

On: UPnP is enabled. The video system advertises its presence until the timeout period expires.

Example: `xConfiguration NetworkServices UPnP Mode: On`

xConfiguration NetworkServices UPnP Timeout

Define for how many seconds UPnP shall stay enabled after the video system is switched on or restarted. The NetworkServices UPnP Mode setting must be On for this setting to take any effect.

Requires user role: ADMIN

Value space: <0..3600>

Range: Select a value between 0 and 3600 seconds.

Example: `xConfiguration NetworkServices UPnP Timeout: 600`

Peripherals configuration

xConfiguration Peripherals Pairing CiscoTouchPanels RemotePairing

In order to use Cisco Touch 10 (touch panel) as user interface for the video system, Touch 10 must be either directly connected to the video system or paired to the video system via LAN. The latter is referred to as remote pairing.

Remote pairing is allowed by default; you must switch this setting Off if you want to prevent remote pairing.

Requires user role: ADMIN

Value space: <Off/On>

Off: Remote pairing of Touch 10 is not allowed.

On: Remote pairing of Touch 10 is allowed.

Example: `xConfiguration Peripherals Pairing CiscoTouchPanels RemotePairing: On`

xConfiguration Peripherals Profile TouchPanels

Set the number of touch panels that are expected to be connected to the video system. This information is used by the video system's diagnostics service. If the number of connected touch panels does not match this setting, the diagnostics service will report it as an inconsistency. Note that only one Cisco Touch controller is supported in this version.

Requires user role: ADMIN

Value space: <NotSet/Minimum1/0/1/2/3/4/5>

NotSet: No touch panel check is performed.

Minimum1: At least one touch panel should be connected to the video system.

0-5: This number of Touch controllers should be connected to the video system.

Example: `xConfiguration Peripherals Profile TouchPanels: NotSet`

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: Not applicable. Removed as from TC7.3.3.

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/Edge>

Off: The video system will not be configured by a provisioning system.

Auto: The provisioning server will automatically be selected by the video system.

TMS: The video system will be configured using TMS (Cisco TelePresence Management System).

VCS: The video system will be configured using VCS (Cisco TelePresence Video Communication Server).

Callway: Not applicable. Removed as from TC7.3.3.

CUCM: The video system will be configured using CUCM (Cisco Unified Communications Manager).

Edge: The system will connect to CUCM via the Collaboration Edge infrastructure.

Example: `xConfiguration Provisioning Mode: Auto`

xConfiguration Provisioning LoginName

This is the user name part of the credentials used to authenticate the video system with the provi-
sioning server. This setting must be used when required by the provisioning server.

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 provi-
sioning server. This setting must be used when required by the provisioning server.

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 AlternateAddress

Only applicable when the endpoint is provisioned by Cisco Unified Communication Manager
(CUCM) and an alternate CUCM is available for redundancy. Enter the address of the alternate
CUCM. If the main CUCM is not available, the endpoint will be provisioned by the alternate CUCM.
When the main CUCM is available again, the endpoint will be provisioned by this CUCM.

Requires user role: ADMIN

Value space: <S: 0, 64>

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

Example: `xConfiguration Provisioning ExternalManager AlternateAddress: ""`

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.

As default, the system is using the UDP ports in the range 2326 to 2487 for RTP and RTCP media data. Each media channel is using two adjacent ports for RTP and RTCP. The default number of ports required in the UDP port range is based on the number of simultaneous calls that the endpoint is capable of.

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

Requires user role: ADMIN

Value space: <1024..65438>
Range: Select a value between 1024 and 65438.

Example: `xConfiguration RTP Ports Range Start: 2326`

xConfiguration RTP Ports Range Stop

Specify the last RTP port in the range.

As default, the system is using the UDP ports in the range 2326 to 2487 for RTP and RTCP media data. Each media channel is using two adjacent ports for RTP and RTCP. The default number of ports required in the UDP port range is based on the number of simultaneous calls that the endpoint is capable of.

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

Requires user role: ADMIN

Value space: <1120..65535>
Range: Select a value between 1120 and 65535.

Example: `xConfiguration RTP Ports Range Stop: 2486`

Security configuration

xConfiguration Security Audit Logging Mode

Determine where to record or transmit the audit logs. The audit logs are sent to a syslog server.

When using the External/ExternalSecure modes and setting the port assignment to manual in the Security Audit Server PortAssignment setting, you must also enter the address and port number for the audit server in the Security Audit Server Address and Security Audit Server Port settings.

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 syslog server. The syslog server must support UDP.

ExternalSecure: The system sends encrypted audit logs to an external 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 syslog server, and the secure TCP server must be set up to listen for secure (TLS) TCP Syslog messages.

Example: `xConfiguration Security Audit Logging Mode: Off`

xConfiguration Security Audit OnError Action

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

Requires user role: AUDIT

Value space: <Halt/Ignore>

Halt: If a halt condition is detected the system codec 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 syslog server. Halt conditions are: A network breach (no physical link), no syslog server running (or incorrect address or port to the syslog server), 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 syslog server.

Example: `xConfiguration Security Audit OnError Action: Ignore`

xConfiguration Security Audit Server Address

The audit logs are sent to a syslog server. Enter the IP address of the syslog 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.

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

The audit logs are sent to a syslog server. Enter the port of the syslog server that the system shall send its audit logs to. This setting is only relevant when Security Audit PortAssignment is set to Manual.

Requires user role: AUDIT

Value space: <0..65535>

Range: Select a value between 0 to 65535.

Example: `xConfiguration Security Audit Server Port: 514`

xConfiguration Security Audit Server PortAssignment

The audit logs are sent to a syslog server. You can define how the port number of the external syslog server will be assigned. This setting is only relevant when Security Audit Logging Mode is set to External or ExternalSecure. To see which port number is used you can check the Security Audit Server Port status. Navigate to Configuration > System status on the web interface or; if on a command line interface, run the command xStatus Security Audit Server Port.

Requires user role: AUDIT

Value space: <Auto/Manual>

Auto: Will use UDP port number 514 when the Security Audit Logging Mode is set to External. Will use TCP port number 6514 when the Security Audit Logging Mode is set to ExternalSecure.

Manual: Will use the port value defined in the Security Audit Server Port setting.

Example: `xConfiguration Security Audit Server PortAssignment: Auto`

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

63

Copyright © 2015 Cisco Systems, Inc. All rights reserved.

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 (COM port).

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 115200.

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: 115200

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 ANAT

ANAT (Alternative Network Address Types) enables media negotiation for multiple addresses and address types, as specified in RFC 4091.

Requires user role: ADMIN

Value space: <Off/On>

Off: Disable ANAT.

On: Enable ANAT.

Example: `xConfiguration SIP ANAT: Off`

xConfiguration SIP AuthenticateTransferror

Not applicable in this version.

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: <Off/On>

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

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

Example: `xConfiguration SIP ListenPort: On`

xConfiguration SIP PreferredIPMedia

Define the preferred IP version for sending and receiving media (audio, video, data). Only applicable when both Network IPStack and Conference CallProtocolIPStack are set to Dual, and the network does not have a mechanism for choosing the preferred IP version.

Requires user role: ADMIN

Value space: <IPv4/IPv6>

IPv4: The preferred IP version for media is IPv4.

IPv6: The preferred IP version for media is IPv6.

Example: `xConfiguration SIP PreferredIPMedia: IPv4`

xConfiguration SIP PreferredIPSignaling

Define the preferred IP version for signaling (audio, video, data). Only applicable when both Network IPStack and Conference CallProtocolIPStack are set to Dual, and the network does not have a mechanism for choosing the preferred IP version. It also determines the priority of the A/AAAA lookups in DNS, so that the preferred IP version is used for registration.

Requires user role: ADMIN

Value space: <IPv4/IPv6>

IPv4: The preferred IP version for signaling is IPv4.

IPv6: The preferred IP version for signaling is IPv6.

Example: `xConfiguration SIP PreferredIPSignaling: IPv4`

xConfiguration SIP OCSP Mode

Not applicable in this version.

xConfiguration SIP OCSP DefaultResponder

Not applicable in this version.

xConfiguration SIP Profile [1..1] Ice Mode

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: <Auto/Off/On>

Auto: When set to Auto, ICE will be enabled if a turn server is provided, otherwise ICE will be disabled.

Off: Set to Off to disable ICE.

On: Set to On to enable ICE.

Example: `xConfiguration SIP Profile 1 Ice Mode: Auto`

xConfiguration SIP Profile [1..1] Ice DefaultCandidate

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

Requires user role: ADMIN

Value space: <Host/Rflx/Relay>

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 Ice DefaultCandidate: Host`

xConfiguration SIP Profile [1..1] Turn DiscoverMode

Set the discover mode to enable/disable the application to search for available Turn servers in DNS. Before making calls, the system will test if port allocation is possible.

Requires user role: ADMIN

Value space: <Off/On>

Off: Set to Off to disable discovery mode.

On: When set to On, the system will search for available Turn servers in DNS, and before making calls the system will test if port allocation is possible.

Example: `xConfiguration SIP Profile Turn DiscoverMode: On`

xConfiguration SIP Profile [1..1] Turn BandwidthProbe

Not applicable in this version.

xConfiguration SIP Profile [1..1] Turn DropRflx

DropRflx will make the endpoint force media through the Turn relay, unless the remote endpoint is on the same network.

Requires user role: ADMIN

Value space: <Off/On>

Off: Disable DropRflx.

On: The system will force media through the Turn relay when the remote endpoint is on another network.

Example: `xConfiguration SIP Profile Turn DropRflx: Off`

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. A random proxy is selected from the list for each SIP outbound request.

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. 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: If SIP Profile Outbound is enabled, use a fully qualified domain name. If SIP Profile Outbound is disabled, you can also use a valid IPv4 address or IPv6 address.

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 behavior for a vendor or provider.

Requires user role: ADMIN

Value space: <Standard/Cisco>

Standard: Use this when registering to standard SIP Proxy (tested with Cisco TelePresence VCS and Broadsoft)

Cisco: Use this when registering to Cisco Unified Communication Manager.

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

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. 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. Requires the Standby Control to be enabled.

Requires user role: ADMIN

Value space: <1..480>

Range: Select a value between 1 and 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

Define the system name. The system name will be sent as the hostname in a DHCP request and when the codec is acting as an SNMP Agent.Define the system name. The system name will be sent as the hostname in a DHCP request and when the codec is acting as an SNMP Agent.

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

This has been replaced with the UserInterface Language setting.

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 xCommand CallHistory Get command.

Requires user role: ADMIN

Value space: <Off/On>

Off: Disable logging.

On: Enable logging.

Example: xConfiguration SystemUnit CallLogging Mode: On

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 IrSensor

Not applicable in this version.

Time configuration

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`

xConfiguration Time Zone

This has been replaced with the Time OlsonZone setting as of software version TC7.2.

xConfiguration Time OlsonZone

Set the time zone for the geographical location of the video system. The information in the value space is from the tz database, also called the IANA Time Zone Database.

Requires user role: USER

Value space: <Africa/Abidjan, Africa/Accra, Africa/Addis_Ababa, Africa/Algiers, Africa/Asmara, Africa/Asmera, Africa/Bamako, Africa/Bangui, Africa/Banjul, Africa/Bissau, Africa/Blantyre, Africa/Brazzaville, Africa/Bujumbura, Africa/Cairo, Africa/Casablanca, Africa/Ceuta, Africa/Conakry, Africa/Dakar, Africa/Dar_es_Salaam, Africa/Djibouti, Africa/Douala, Africa/El_Aaiun, Africa/Freetown, Africa/Gaborone, Africa/Harare, Africa/Johannesburg, Africa/Juba, Africa/Kampala, Africa/Khartoum, Africa/Kigali, Africa/Kinshasa, Africa/Lagos, Africa/Libreville, Africa/Lome, Africa/Luanda, Africa/Lubumbashi, Africa/Lusaka, Africa/Malabo, Africa/Maputo, Africa/Maseru, Africa/Mbabane, Africa/Mogadishu, Africa/Monrovia, Africa/Nairobi, Africa/Ndjamena, Africa/Niamey, Africa/Nouakchott, Africa/Ouagadougou, Africa/Porto-Novo, Africa/Sao_Tome, Africa/Timbuktu, Africa/Tripoli, Africa/Tunis, Africa/Windhoek, America/Adak, America/Anchorage, America/Anguilla, America/Antigua, America/Araguaina, America/Argentina/Buenos_Aires, America/Argentina/Catamarca, America/Argentina/ComodRivadavia, America/Argentina/Cordoba, America/Argentina/Jujuy, America/Argentina/La_Rioja, America/Argentina/Mendoza, America/Argentina/Rio_Gallegos, America/Argentina/Salta, America/Argentina/San_Juan, America/Argentina/San_Luis, America/Argentina/Tucuman, America/Argentina/Ushuaia, America/Aruba, America/Asuncion, America/Atikokan, America/Atka, America/Bahia, America/Bahia_Banderas, America/Barbados, America/Belem, America/Belize, America/Blanc-Sablon, America/Boa_Vista, America/Bogota, America/Boise, America/Buenos_Aires, America/Cambridge_Bay, America/Campo_Grande, America/Cancun, America/Caracas, America/Catamarca, America/Cayenne, America/Cayman, America/Chicago, America/Chihuahua, America/Coral_Harbour, America/Cordoba, America/Costa_Rica, America/Creston, America/Cuiaba, America/Curacao, America/Danmarkshavn, America/Dawson, America/Dawson_Creek, America/Denver, America/Detroit, America/Dominica, America/Edmonton, America/Eirunepe, America/El_Salvador, America/Ensenada, America/Fort_Wayne, America/Fortaleza, America/Glace_Bay, America/Godthab, America/Goose_Bay, America/Grand_Turk, America/Grenada, America/Guadeloupe, America/Guatemala, America/Guayaquil, America/Guyana, America/Halifax, America/Havana, America/Hermosillo, America/Indiana/Indianapolis, America/Indiana/Knox, America/Indiana/Marengo, America/Indiana/Petersburg, America/Indiana/Tell_City, America/Indiana/Vevay, America/Indiana/Vincennes, America/Indiana/Winamac, America/Indianapolis, America/Inuvik, America/Iqaluit, America/Jamaica, America/Jujuy, America/Juneau, America/Kentucky/Louisville, America/Kentucky/Monticello, America/Knox_IN, America/Kralendijk, America/La_Paz, America/Lima, America/Los_Angeles, America/Louisville, America/Lower_Princes, America/Maceio, America/Managua, America/Manaus, America/Marigot, America/Martinique, America/Matamoros, America/Mazatlan, America/Mendoza, America/Menominee, America/Merida, America/Metlakatla, America/Mexico_City, America/Miquelon, America/Moncton, America/Monterrey, America/Montevideo, America/Montreal, America/Montserrat, America/Nassau, America/New_York, America/Nipigon, America/Nome, America/Noronha, America/North_Dakota/Beulah, America/North_Dakota/Center, America/North_Dakota/New_Salem, America/Ojinaga, America/Panama, America/Pangnirtung, America/Paramaribo, America/Phoenix, America/Port-au-Prince, America/Port_of_Spain, America/Porto_Acre, America/Porto_Velho, America/Puerto_Rico, America/Rainy_River, America/Rankin_Inlet, America/Recife, America/Regina, America/Resolute, America/Rio_Branco, America/Rosario, America/Santa_Isabel, America/Santarem, America/Santiago, America/Santo_Domingo, America/Sao_Paulo, America/Scoresbysund, America/Shiprock, America/Sitka, America/St_Barthelemy, America/St_Johns, America/St_Kitts, America/St_Lucia, America/St_Thomas, America/St_Vincent, America/

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

71

Swift_Current, America/Tegucigalpa, America/Thule, America/Thunder_Bay, America/Tijuana, America/Toronto, America/Tortola, America/Vancouver, America/Virgin, America/Whitehorse, America/Winnipeg, America/Yakutat, America/Yellowknife, Antarctica/Casey, Antarctica/Davis, Antarctica/DumontDUrville, Antarctica/Macquarie, Antarctica/Mawson, Antarctica/McMurdo, Antarctica/Palmer, Antarctica/Rothera, Antarctica/South_Pole, Antarctica/Syowa, Antarctica/Vostok, Arctic/Longyearbyen, Asia/Aden, Asia/Almaty, Asia/Amman, Asia/Anadyr, Asia/Aqtau, Asia/Aqtobe, Asia/Ashgabat, Asia/Ashkhabad, Asia/Baghdad, Asia/Bahrain, Asia/Baku, Asia/Bangkok, Asia/Beirut, Asia/Bishkek, Asia/Brunei, Asia/Calcutta, Asia/Choibalsan, Asia/Chongqing, Asia/Chungking, Asia/Colombo, Asia/Dacca, Asia/Damascus, Asia/Dhaka, Asia/Dili, Asia/Dubai, Asia/Dushanbe, Asia/Gaza, Asia/Harbin, Asia/Hebron, Asia/Ho_Chi_Minh, Asia/Hong_Kong, Asia/Hovd, Asia/Irkutsk, Asia/Istanbul, Asia/Jakarta, Asia/Jayapura, Asia/Jerusalem, Asia/Kabul, Asia/Kamchatka, Asia/Karachi, Asia/Kashgar, Asia/Kathmandu, Asia/Katmandu, Asia/Khandyga, Asia/Kolkata, Asia/Krasnoyarsk, Asia/Kuala_Lumpur, Asia/Kuching, Asia/Kuwait, Asia/Macao, Asia/Macau, Asia/Magadan, Asia/Makassar, Asia/Manila, Asia/Muscat, Asia/Nicosia, Asia/Novokuznetsk, Asia/Novosibirsk, Asia/Omsk, Asia/Oral, Asia/Phnom_Penh, Asia/Pontianak, Asia/Pyongyang, Asia/Qatar, Asia/Qyzylorda, Asia/Rangoon, Asia/Riyadh, Asia/Saigon, Asia/Sakhalin, Asia/Samarkand, Asia/Seoul, Asia/Shanghai, Asia/Singapore, Asia/Taipei, Asia/Tashkent, Asia/Tbilisi, Asia/Tehran, Asia/Tel_Aviv, Asia/Thimbu, Asia/Thimphu, Asia/Tokyo, Asia/Ujung_Pandang, Asia/Ulaanbaatar, Asia/Ulan_Bator, Asia/Urumqi, Asia/Ust-Nera, Asia/Vientiane, Asia/Vladivostok, Asia/Yakutsk, Asia/Yekaterinburg, Asia/Yerevan, Atlantic/Azores, Atlantic/Bermuda, Atlantic/Canary, Atlantic/Cape_Verde, Atlantic/Faeroe, Atlantic/Faroe, Atlantic/Jan_Mayen, Atlantic/Madeira, Atlantic/Reykjavik, Atlantic/South_Georgia, Atlantic/St_Helena, Atlantic/Stanley, Australia/ACT, Australia/Adelaide, Australia/Brisbane, Australia/Broken_Hill, Australia/Canberra, Australia/Currie, Australia/Darwin, Australia/Eucla, Australia/Hobart, Australia/LHI, Australia/Lindeman, Australia/Lord_Howe, Australia/Melbourne, Australia/NSW, Australia/North, Australia/Perth, Australia/Queensland, Australia/South, Australia/Sydney, Australia/Tasmania, Australia/Victoria, Australia/West, Australia/Yancowinna, Brazil/Acre, Brazil/DeNoronha, Brazil/East, Brazil/West, CET, CST6CDT, Canada/Atlantic, Canada/Central, Canada/East-Saskatchewan, Canada/Eastern, Canada/Mountain, Canada/Newfoundland, Canada/Pacific, Canada/Saskatchewan, Canada/Yukon, Chile/Continental, Chile/EasterIsland, Cuba, EET, EST, EST5EDT, Egypt, Eire, Etc/GMT, Etc/GMT+0, Etc/GMT+1, Etc/GMT+10, Etc/GMT+11, Etc/GMT+12, Etc/GMT+2, Etc/GMT+3, Etc/GMT+4, Etc/GMT+5, Etc/GMT+6, Etc/GMT+7, Etc/GMT+8, Etc/GMT+9, Etc/GMT-0, Etc/GMT-1, Etc/GMT-10, Etc/GMT-11, Etc/GMT-12, Etc/GMT-13, Etc/GMT-14, Etc/GMT-2, Etc/GMT-3, Etc/GMT-4, Etc/GMT-5, Etc/GMT-6, Etc/GMT-7, Etc/GMT-8, Etc/GMT-9, Etc/GMT0, Etc/Greenwich, Etc/UCT, Etc/UTC, Etc/Universal, Etc/Zulu, Europe/Amsterdam, Europe/Andorra, Europe/Athens, Europe/Belfast, Europe/Belgrade, Europe/Berlin, Europe/Bratislava, Europe/Brussels, Europe/Bucharest, Europe/Budapest, Europe/Busingen, Europe/Chisinau, Europe/Copenhagen, Europe/Dublin, Europe/Gibraltar, Europe/Guernsey, Europe/Helsinki, Europe/Isle_of_Man, Europe/Istanbul, Europe/Jersey, Europe/Kaliningrad, Europe/Kiev, Europe/Lisbon, Europe/Ljubljana, Europe/London, Europe/Luxembourg, Europe/Madrid, Europe/Malta, Europe/Mariehamn, Europe/Minsk, Europe/Monaco, Europe/Moscow, Europe/Nicosia, Europe/Oslo, Europe/Paris, Europe/Podgorica, Europe/Prague, Europe/Riga, Europe/Rome, Europe/Samara, Europe/San_Marino, Europe/Sarajevo, Europe/Simferopol, Europe/Skopje, Europe/Sofia, Europe/Stockholm, Europe/Tallinn, Europe/Tirane, Europe/Tiraspol, Europe/Uzhgorod, Europe/Vaduz, Europe/Vatican, Europe/Vienna, Europe/Vilnius, Europe/Volgograd, Europe/Warsaw, Europe/Zagreb, Europe/Zaporozhye, Europe/Zurich, GB, GB-Eire, GMT, GMT+0, GMT-0, GMT0, Greenwich, HST, Hongkong, Iceland, Indian/Antananarivo, Indian/Chagos, Indian/Christmas, Indian/Cocos, Indian/Comoro, Indian/Kerguelen, Indian/Mahe, Indian/Maldives, Indian/Mauritius, Indian/Mayotte, Indian/Reunion, Iran, Israel, Jamaica, Japan, Kwajalein, Libya, MET, MST, MST7MDT, Mexico/BajaNorte, Mexico/BajaSur, Mexico/General, NZ, NZ-CHAT, Navajo, PRC, PST8PDT, Pacific/Apia, Pacific/Auckland, Pacific/Chatham, Pacific/Chuuk,

Pacific/Easter, Pacific/Efate, Pacific/Enderbury, Pacific/Fakaofo, Pacific/Fiji, Pacific/Funafuti, Pacific/Galapagos, Pacific/Gambier, Pacific/Guadalcanal, Pacific/Guam, Pacific/Honolulu, Pacific/Johnston, Pacific/Kiritimati, Pacific/Kosrae, Pacific/Kwajalein, Pacific/Majuro, Pacific/Marquesas, Pacific/Midway, Pacific/Nauru, Pacific/Niue, Pacific/Norfolk, Pacific/Noumea, Pacific/Pago_Pago, Pacific/Palau, Pacific/Pitcairn, Pacific/Pohnpei, Pacific/Ponape, Pacific/Port_Moresby, Pacific/Rarotonga, Pacific/Saipan, Pacific/Samoa, Pacific/Tahiti, Pacific/Tarawa, Pacific/Tongatapu, Pacific/Truk, Pacific/Wake, Pacific/Wallis, Pacific/Yap, Poland, Portugal, ROC, ROK, Singapore, Turkey, UCT, US/Alaska, US/Aleutian, US/Arizona, US/Central, US/East-Indiana, US/Eastern, US/Hawaii, US/Indiana-Starke, US/Michigan, US/Mountain, US/Pacific, US/Pacific-New, US/Samoa, UTC, Universal, W-SU, WET, Zulu>

Range: Select a time zone from the list.

Example: `xConfiguration Time OlsonZone: Etc/UTC`

72

UserInterface configuration

xConfiguration UserInterface Language

Select the language to be used in menus and messages on the screen and Touch controller. The default language is English.

Requires user role: USER

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

Range: Select a language from the list.

Example: `xConfiguration UserInterface Language: English`

xConfiguration UserInterface 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 UserInterface OSD EncryptionIndicator: Auto`

xConfiguration UserInterface OSD LanguageSelection

In cases where you want to prevent users from easily changing the language settings from the Settings menu, the language settings can be made available from within the Administrator Settings menu. The administrator settings can be password protected.

Requires user role: ADMIN

Value space: <Off/On>

Off: The language is set from the Administrator Settings menu.

On: The language is set from the Settings menu.

Example: `xConfiguration UserInterface OSD LanguageSelection: On`

xConfiguration UserInterface OSD LoginRequired

Not applicable in this version.

xConfiguration UserInterface OSD Output

Define on which monitor the on-screen information and indicators should be displayed.

Requires user role: ADMIN

Value space: <Auto/1/2/3>

Auto: The system will detect when a monitor is connected to a video output, and send the information and indicators to the first monitor you connect. If you have a multi-monitor setup, and all monitors are connected before switching on the system, the information and indicators will be sent to the video output with the lowest number, starting with Output Connector 1 (HDMI 1).

Range 1-3: The system will send the on-screen information and indicators to the specified output. Choose n to send the information and indicators to the system's Output Connector n.

Example: `xConfiguration UserInterface OSD Output: Auto`

xConfiguration UserInterface Wallpaper

Select a background image (wallpaper) for the video screen when idle.

You may upload a custom wallpaper to the video system using the web interface. The following file formats are supported: BMP, GIF, JPEG, PNG. The maximum file size is 2 MByte.

Requires user role: USER

Value space: <None/Custom>

None: There is no background image on the screen.

Custom: Use the custom wallpaper as background image on the screen. If no custom wallpaper is uploaded to the system, the setting will revert to the default value.

Example: `xConfiguration UserInterface Wallpaper: None`

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

73

Copyright © 2015 Cisco Systems, Inc. All rights reserved.

xConfiguration UserInterface TouchPanel DefaultPanel

Define what (contact list, meeting list, or dial pad) the Touch controller will display on wake up.

Requires user role: USER

Value space: <None/LastUsed/ContactList/MeetingList/Dialpad>

None: None of the below options will appear as default on the Touch controller.

LastUsed: The last used (contact list, meeting list, or dial pad) will appear as default on the Touch controller.

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

Example: xConfiguration UserInterface TouchPanel DefaultPanel: None

xConfiguration UserInterface UserPreferences

Some user preferences (ringtone, volume, language, date and time, etc) can be made available from the Settings menu, or from the Settings > Administrator menu on the Touch controller. Accessing the Administrator menus requires that the user has admin privileges.

Requires user role: ADMIN

Value space: <Off/On>

Off: The user preferences are available from the Settings > Administrator menu on the Touch controller, for users with admin privileges.

On: The user preferences are available from the Settings menu on the Touch controller.

Example: xConfiguration UserInterface UserPreferences: On

Video configuration

xConfiguration Video AllowWebSnapshots

Note: This setting is only available in TC7.3.0 to TC7.3.2.

Allow or disallow snapshots being taken of the local input sources, remote sites and presentation channel. If snapshots are allowed, the snapshots may be captured both when idle and in a call.

When snapshots are taken from a remote device, e.g. the web interface, a notification appears on the video system's screens to alert the users that remote monitoring is in operation.

Requires user role: ADMIN

Value space: <Off/On/LocalDeviceOnly>

Off: It is not possible to capture snapshots.

On: Snapshots can be captured and displayed anywhere, e.g. on the web interface.

LocalDeviceOnly: Snapshots can only be captured and displayed on devices running the experimental Cisco Proximity feature. The devices must be in the same room as the video system. It will not be possible to take and see snapshots on the web interface or by using 3rd party integrations.

Example: xConfiguration Video AllowWebSnapshots: LocalDeviceOnly

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: On

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 DefaultPresentationSource

Not applicable for this product.

xConfiguration Video Input Connector [1..5] Name

Enter a name for the video input connector.

Requires user role: ADMIN

Value space: <S: 0, 50>

Format: String with a maximum of 50 characters.

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

xConfiguration Video Input Connector [1..5] InputSourceType

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

Requires user role: ADMIN

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

other: Use this when none of the below options match.
camera: Use this when a camera is connected to the video input.
PC: Use this when a computer is connected to the video input.
DVD: Use this when a DVD player is connected to the video input.
document_camera: Use this when a document camera is connected to the video input.
whiteboard: Use this when a whiteboard camera is connected to the video input.

Example: `xConfiguration Video Input Connector 2 InputSourceType: camera`

xConfiguration Video Input Connector [1..5] Visibility

Define the visibility of the video input connector in the menus on the user interface.

Requires user role: ADMIN

Value space: <Never/Always/IfSignal>

Never: When the input source is not expected to be used as a presentation source, set to Never.
Always: When set to Always, the menu selection for the video input connector will always be visible on the graphical user interface.
IfSignal: When set to IfSignal, the menu selection for the video input connector will only be visible when something is connected to the video input.

Example: `xConfiguration Video Input Connector 2 Visibility: IfSignal`

xConfiguration Video Input Connector [1..5] CameraControl Mode

Define whether the camera that is connected to this video input connector can be controlled or not.

Note that camera control is not available for Connector 4 (DVI-I) and Connector 5 (S-video/Composite).

Requires user role: ADMIN

Value space: Connector 1, 2, 3: <Off/On>   Connector 4,5: <Off>

Off: Disable camera control.
On: Enable camera control.

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

xConfiguration Video Input Connector [1..5] CameraControl CameraId

The camera ID is used to identify all cameras that are controlled from the codec. Use the xStatus Camera API command to see the IDs of the different cameras.

Requires user role: ADMIN

Value space: Connector 1, 2, 3: <1/2/3/4/5/6/7>   Connector 4,5: <1>

Range: Select the ID of the camera.

Example: `xConfiguration Video Input Connector 1 CameraControl CameraId: 1`

xConfiguration Video Input Connector [1..5] Quality

When encoding and transmitting video there will be a trade-off between high resolution and high frame rate. For some video sources it is more important to transmit high frame rate than high resolution and vice versa.

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 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 Connector 3 Quality: Sharpness`

xConfiguration Video Input Connector [1..5] OptimalDefinition Profile

This setting will only take effect if the corresponding Video Input Connector Quality setting is set to Motion.

The optimal definition profile reflects the lighting conditions in the video conferencing room and the quality of the camera. The better lighting conditions and the better quality of the camera, the higher the profile. In good lighting conditions, the video encoder will provide better quality (higher resolution or frame rate) for a given call rate. Generally, the Normal or Medium profiles are recommended. However, when the lighting conditions are very 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. 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					
			768 kbps	1152 kbps	1472 kbps	2560 kbps	4 Mbps*	6 Mbps*
H.265	30 fps	Normal	1280×720	1280×720	1280×720	1920×1080	1920×1080	1920×1080
		Medium	1280×720	1920×1080	1920×1080	1920×1080	1920×1080	1920×1080
		High	1920×1080	1920×1080	1920×1080	1920×1080	1920×1080	1920×1080
	60 fps	Normal	768×448	1024×576	1280×720	1280×720	1280×720	1280×720
		Medium	1024×576	1280×720	1280×720	1280×720	1280×720	1280×720
		High	1280×720	1280×720	1280×720	1280×720	1280×720	1280×720
H.264	30 fps	Normal	1024×576	1280×720	1280×720	1920×1080	1920×1080	1920×1080
		Medium	1280×720	1280×720	1280×720	1920×1080	1920×1080	1920×1080
		High	1280×720	1280×720	1920×1080	1920×1080	1920×1080	1920×1080
	60 fps	Normal	640×360	768×448	1024×576	1280×720	1280×720	1920×1080
		Medium	768×448	1024×576	1024×576	1280×720	1920×1080	1920×1080
		High	1024×576	1280×720	1280×720	1920×1080	1920×1080	1920×1080

* H.265 is preferred over H.264, and the maximum bit rate for H.265 is 3 Mbps. When the user sets a higher bit rate, the codec will still use H.265 at 3 Mbps as long as all codecs involved supports H.265.

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 Connector 1 OptimalDefinition Profile: Medium`

xConfiguration Video Input Connector [1..5] 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 frame rate 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 Connector 1 OptimalDefinition Threshold60fps: 1280 _ 720`

xConfiguration Video Input Connector [1..4] PresentationSelection

Define how the video system will behave when you connect a presentation source to the video input.

If the video system is in standby mode, it will wake up when you connect a presentation source. Note that sharing the presentation with the far end always requires additional action (press Share on the user interface).

Requires user role: ADMIN

Value space: <Manual/Automatic/OnConnect>

Manual: In manual mode, the contents of the video input will not be presented on the screen until you choose it from the user interface.

Automatic: In automatic mode, the contents on the video input will be presented on screen automatically. If more than one source is set to Automatic, the last connected source will be used. If any content is active (presented) when a call is disconnected, the content will still be displayed locally.

OnConnect: When in on-connect mode, the content on the video input will be presented on screen when a cable is connected. Otherwise, the behavior is the same as in manual mode.

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

xConfiguration Video Input Connector [1..4] RGBQuantizationRange

The devices connected to the video input 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 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 Connector 1 RGBQuantizationRange: Auto`

xConfiguration Video Input Connector [4] DviType

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 Connector 4 DviType: AutoDetect`

xConfiguration Video Input Connector [5] SignalType

Connector 5 can be used for either S-Video or Composite video input format. Use this setting to configure which video format the BNC connector(s) are used for.

Requires user role: ADMIN

Value space: <Composite/YC>

Composite: Connector 5 is configured for composite video input. Only the BNC connector that is labeled "Y" is used.

YC: Connector 5 is configured for S-Video input. Both BNC connectors ("Y" and "C") are used.

Example: `xConfiguration Video Input Connector 5 SignalType: Composite`

xConfiguration Video Layout DisableDisconnectedLocalOutputs

This setting is fixed to On.

Requires user role: ADMIN

Value space: <On>

On: The built-in layout engine does only set layout on local outputs having a monitor connected.

Example: `xConfiguration Video Layout DisableDisconnectedLocalOutputs: On`

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: Do not use this value.

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: Do not use this value.

PresentationLargeSpeaker: Do not use this value.

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 PresentationDefault View

Determine how the presentation will show on screen when you start sharing a presentation.

Requires user role: ADMIN

Value space: <Default/Minimized/Maximized>

Default: The presentation is a part of the layout.

Minimized: The presentation starts up in PIP mode.

Maximized: The presentation starts up in full screen mode.

Example: `xConfiguration Video Layout PresentationDefault View: Default`

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: Do not use this value.

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: Do not use this value.

PresentationLargeSpeaker: Do not use this value.

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 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 between 0 and 100 percent.

Example: `xConfiguration Video Layout ScaleToFrameThreshold: 5`

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 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 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 SelfviewDefault FullscreenMode settings respectively.

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`

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

79

Copyright © 2015 Cisco Systems, Inc. All rights reserved.

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).

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 Connector [n] MonitorRole setting.

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 or Triple.

Requires user role: ADMIN

Value space: <First/Second/Third/Current>

First: The self-view picture will be shown on outputs with the Video Output Connector [n] MonitorRole set to First.

Second: The self-view picture will be shown on outputs with the Video Output Connector [n] MonitorRole set to Second.

Third: The self-view picture will be shown on outputs with the Video Output Connector [n] MonitorRole set to Third.

Current: When leaving a call, the self-view picture will be kept on the same output as it was during the call.

Example: xConfiguration Video SelfviewDefault OnMonitorRole: Current

xConfiguration Video Monitors

A role is assigned to each monitor using the Video Output Connector [n] MonitorRole setting. The monitor role decides which layout (call participants and presentation) will appear on the monitor that is connected to this output. Monitors with the same monitor role will get the same layout; monitors with different monitor roles will have different layouts.

The monitor layout mode that is set in the Video Monitors setting should reflect the number of different layouts you want in your room setup. Note that some monitors can be reserved for presentations.

Requires user role: ADMIN

Value space: <Auto/Single/Dual/DualPresentationOnly/TriplePresentationOnly/Triple>

Auto: The number of monitors connected to the codec is automatically detected, and the layout is distributed on the monitors according to the MonitorRole settings.

Single: The same layout is shown on all monitors.

Dual: The layout is distributed on monitors with monitor role First and Second. If a presentation is part of the layout, all participants in the call are shown on monitors with monitor role First, and the presentation is shown on monitors with monitor role Second.

DualPresentationOnly: All participants in the call are shown on monitors with monitor role First. If a presentation is part of the layout, the presentation is shown on monitors with monitor role Second.

Triple: The layout is distributed on monitors with monitor role First, Second and Third. If a presentation is part of the layout, all participants in the call are shown on monitors with monitor role First and Second, and the presentation is shown on the monitor with monitor role Third.

TriplePresentationOnly: All participants in the call are distributed on monitors with monitor role First and Second. If a presentation is part of the layout, the presentation is shown on the monitor with monitor role Third.

Example: xConfiguration Video Monitors: Auto

xConfiguration Video OSD LanguageSelection

This has been replaced with the UserInterface OSD LanguageSelection setting.

xConfiguration Video OSD EncryptionIndicator

This has been replaced with the UserInterface OSD EncryptionIndicator setting.

xConfiguration Video OSD Output

This has been replaced with the UserInterface OSD Output setting.

xConfiguration Video OSD LoginRequired

This has been replaced with the UserInterface OSD LoginRequired setting.

xConfiguration Video Output Connector [1..2] CEC Mode

This video output (HDMI) supports Consumer Electronics Control (CEC). When this setting is On (default is Off), the system will use CEC to set the monitor in standby when the system itself enters standby. Likewise the system will wake up the monitor when the system itself wakes up from standby. For this to happen, the monitor that is connected to the output must be CEC compatible and CEC must be configured on the monitor.

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 Connector 1 CEC Mode: Off

xConfiguration Video Output Connector [1..3] Location HorizontalOffset

HorizontalOffset and VerticalOffset settings are associated with each video output. These settings are used to signal the relative position of the displays that are connected to these outputs.

HorizontalOffset = 0 and VerticalOffset = 0 indicates that the display is positioned in center, both horizontally and vertically. A negative horizontal offset indicates that the monitor is left of center, and a positive horizontal offset indicates that the monitor is right of center. A negative vertical offset indicates that the monitor is below center, and a positive vertical offset indicates that the monitor is above center. The magnitude of the offset indicates how far the display is from center (relative to other displays).

Example: You have three displays side by side, with the left and right displays at equal distance from center. Then the following settings will apply: HorizontalOffset = 0 for the center display, HorizontalOffset = -1 for the left display, and HorizontalOffset = 1 for the right display.

Example: You have two displays, one in center and one below. Then the following settings will apply: VerticalOffset = 0 for the center display, Vertical Offset = -1 for the lower display.

The default values for the different outputs are:

Video Output Connector [1] Location: HorizontalOffset = -1, VerticalOffset = 0
Video Output Connector [2] Location: HorizontalOffset = 0, VerticalOffset = 0
Video Output Connector [3] Location: HorizontalOffset = 1, VerticalOffset = 0

Requires user role: ADMIN

Value space: <-100..100>

Range: The value must be between -100 and 100.

Example: xConfiguration Video Output Connector 2 Location HorizontalOffset: -1

xConfiguration Video Output Connector [1..3] Location VerticalOffset

HorizontalOffset and VerticalOffset settings are associated with each video output. These settings are used to signal the relative position of the displays that are connected to these outputs.

HorizontalOffset = 0 and VerticalOffset = 0 indicates that the display is positioned in center, both horizontally and vertically. A negative horizontal offset indicates that the monitor is left of center, and a positive horizontal offset indicates that the monitor is right of center. A negative vertical offset indicates that the monitor is below center, and a positive vertical offset indicates that the monitor is above center. The magnitude of the offset indicates how far the display is from center (relative to other displays).

Example: You have three displays side by side, with the left and right displays at equal distance from center. Then the following settings will apply: HorizontalOffset = 0 for the center display, HorizontalOffset = -1 for the left display, and HorizontalOffset = 1 for the right display.

Example: You have two displays, one in center and one below. Then the following settings will apply: VerticalOffset = 0 for the center display, Vertical Offset = -1 for the lower display.

The default values for the different outputs are:

Video Output Connector [1] Location: HorizontalOffset = -1, VerticalOffset = 0
Video Output Connector [2] Location: HorizontalOffset = 0, VerticalOffset = 0
Video Output Connector [3] Location: HorizontalOffset = 1, VerticalOffset = 0

Requires user role: ADMIN

Value space: <-100..100>

Range: The value must be between -100 and 100.

Example: xConfiguration Video Output Connector 2 Location Vertical Offset: 0

xConfiguration Video Output Connector [1..3] RGBQuantizatonRange

Devices connected to an HDMI output 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 Connector 1 RGBQuantizatonRange: Full

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

81

Copyright © 2015 Cisco Systems, Inc. All rights reserved.

xConfiguration Video Output Connector [1..3] Resolution

Set the resolution and refresh rate for the connected screen.

Requires user role: ADMIN

Value space: <Auto/1280_720_50/1280_720_60/1920_1080_50/1920_1080_60>

Auto: The system will automatically try to set the optimal resolution based on negotiation with the connected monitor.

1280_720_50: The resolution is 1280 x 720, and the refresh rate is 50 Hz.

1280_720_60: The resolution is 1280 x 720, and the refresh rate is 60 Hz.

1920_1080_50: The resolution is 1920 x 1080, and the refresh rate is 50 Hz.

1920_1080_60: The resolution is 1920 x 1080, and the refresh rate is 60 Hz.

Example: `xConfiguration Video Output Connector 2 Resolution: Auto`

xConfiguration Video Output Connector [1..3] MonitorRole

The monitor role describes which video streams will be shown on the monitor connected to this video output connector. Together the Video Monitors setting and the MonitorRole settings for all outputs define which layout (video streams) will be shown on each monitor.

Requires user role: ADMIN

Value space: <Auto/First/Second/Third/PresentationOnly/Recorder>

Auto: The system will detect when a monitor is connected, and a monitor role (First, Second, Third) that corresponds with the Video Monitors setting will be assigned automatically.

First/Second/Third: Define the role of the monitor in a multi-monitor setup. In a single-monitor setup, there is no difference between First, Second and Third.

PresentationOnly: Show presentation video stream if active, and nothing else. Monitors/outputs with this monitor role are disregarded by the Video Monitors setting.

Recorder: Show all participants, including the local main video (self-view). If active, also show the presentation. Monitors/outputs with this monitor role are disregarded by the Video Monitors setting.

Example: `xConfiguration Video Output Connector 1 MonitorRole: Auto`

xConfiguration Video WallPaper

This has been replaced with the UserInterface Wallpaper setting.

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

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

83

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/sx-docs

Audio commands

xCommand Audio Equalizer List

Shows the current equalizer settings for the codec.

Requires user role: ADMIN

Parameters:

EqualizerId: <1..8>

Example:

```
xCommand Audio Equalizer List EqualizerId: 1
*r AudioEqualizerListResult Equalizer 1 Section 1 FilterType: "Peaking"
*r AudioEqualizerListResult Equalizer 1 Section 1 Frequency: "100"
*r AudioEqualizerListResult Equalizer 1 Section 1 Q: "4"
*r AudioEqualizerListResult Equalizer 1 Section 1 Gain: "0"
*r AudioEqualizerListResult Equalizer 1 Section 2 FilterType: "Peaking"
*r AudioEqualizerListResult Equalizer 1 Section 2 Frequency: "200"
*r AudioEqualizerListResult Equalizer 1 Section 2 Q: "4"
*r AudioEqualizerListResult Equalizer 1 Section 2 Gain: "0"
*r AudioEqualizerListResult Equalizer 1 Section 3 FilterType: "Peaking"
*r AudioEqualizerListResult Equalizer 1 Section 3 Frequency: "500"
*r AudioEqualizerListResult Equalizer 1 Section 3 Q: "4"
*r AudioEqualizerListResult Equalizer 1 Section 3 Gain: "0"
*r AudioEqualizerListResult Equalizer 1 Section 4 FilterType: "Peaking"
*r AudioEqualizerListResult Equalizer 1 Section 4 Frequency: "1000"
*r AudioEqualizerListResult Equalizer 1 Section 4 Q: "4"
*r AudioEqualizerListResult Equalizer 1 Section 4 Gain: "0"
*r AudioEqualizerListResult Equalizer 1 Section 5 FilterType: "Peaking"
*r AudioEqualizerListResult Equalizer 1 Section 5 Frequency: "2000"
*r AudioEqualizerListResult Equalizer 1 Section 5 Q: "4"
*r AudioEqualizerListResult Equalizer 1 Section 5 Gain: "0"
*r AudioEqualizerListResult Equalizer 1 Section 6 FilterType: "Peaking"
*r AudioEqualizerListResult Equalizer 1 Section 6 Frequency: "5000"
*r AudioEqualizerListResult Equalizer 1 Section 6 Q: "0"
*r AudioEqualizerListResult Equalizer 1 Section 6 Gain: "0"
```

xCommand Audio Equalizer Update

The system has eight user defined equalizers, each made up of six second order IIR sections. Each of the eight equalizers can be applied to one or more of the audio input and output connectors on the codec. Each IIR section can be modified independently.

There are five filter types, and the frequency response variations dependent on some of the parameter variations. The Q-value for low pass, high pass, low shelf and high shelf filters should be set to 1/sqrt(2) in order to get maximally flat responses. The Q-value (or Q-factor) is defined as Q=f0/bw. Where f0=resonance frequency in Hz; and bw=filter bandwidth in Hz.

To switch off one of the six equalizer sections; set the second order section to have a flat frequency response. This can be done by setting the filter type to "none" or by setting the filter type to "peaking" and the gain to "0" (zero).

We recommend using the TC Console tool, with the embedded equalizer GUI, to modify the equalizers. The TC Console software is found on the Developer Zone web page. Go to: http://developer.cisco.com/web/telepresence-developer/tools/integrators/tc-console.

Requires user role: ADMIN

Parameters:

EqualizerId(r): <1..8>
Section(r): <1..6>
FilterType(r): <highpass/highshelf/lowpass/lowshelf/none/peaking>
Frequency(r): <S: 0, 32>
Q(r): <S: 0, 32>
Gain(r): <S: 0, 32>

Example:

```
xCommand Audio Equalizer List EqualizerId: 1 Section: 1 FilterType:
Peaking Frequency: "100" Q: "4" Gain: "0"
*r AudioEqualizerListResult Equalizer 1 Section 1 FilterType: "Peaking"
*r AudioEqualizerListResult Equalizer 1 Section 1 Frequency: "100"
*r AudioEqualizerListResult Equalizer 1 Section 1 Q: "4"
*r AudioEqualizerListResult Equalizer 1 Section 1 Gain: "0"
*r AudioEqualizerListResult Equalizer 1 Section 2 FilterType: "Peaking"
*r AudioEqualizerListResult Equalizer 1 Section 2 Frequency: "200"
*r AudioEqualizerListResult Equalizer 1 Section 2 Q: "4"
*r AudioEqualizerListResult Equalizer 1 Section 2 Gain: "0"
*r AudioEqualizerListResult Equalizer 1 Section 3 FilterType: "Peaking"
*r AudioEqualizerListResult Equalizer 1 Section 3 Frequency: "500"
*r AudioEqualizerListResult Equalizer 1 Section 3 Q: "4"
*r AudioEqualizerListResult Equalizer 1 Section 3 Gain: "0"
*r AudioEqualizerListResult Equalizer 1 Section 4 FilterType: "Peaking"
*r AudioEqualizerListResult Equalizer 1 Section 4 Frequency: "1000"
```

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

88

Copyright © 2015 Cisco Systems, Inc. All rights reserved.

```
*r AudioEqualizerListResult Equalizer 1 Section 4 Q: "4"
*r AudioEqualizerListResult Equalizer 1 Section 4 Gain: "0"
*r AudioEqualizerListResult Equalizer 1 Section 5 FilterType: "Peaking"
*r AudioEqualizerListResult Equalizer 1 Section 5 Frequency: "2000"
*r AudioEqualizerListResult Equalizer 1 Section 5 Q: "4"
*r AudioEqualizerListResult Equalizer 1 Section 5 Gain: "0"
*r AudioEqualizerListResult Equalizer 1 Section 6 FilterType: "Peaking"
*r AudioEqualizerListResult Equalizer 1 Section 6 Frequency: "5000"
*r AudioEqualizerListResult Equalizer 1 Section 6 Q: "0"
*r AudioEqualizerListResult Equalizer 1 Section 6 Gain: "0"
```

xCommand Audio LocalInput Add

Create a local input and generate the local input id. A local input is a mix of input connectors with the following settings: Name, MixerMode, AGC, Mute and Channels.

This command is mainly intended for internal use by the TC console. TC console automatically generates a start-up script that makes the command persistent over a reboot. To use this command outside of TC console, you have to include it in a start-up script in order to make the setting persistent over a reboot. For more information on star-up scripts and TC console see the administrator guide for your codec.

InputId: A unique identifier for the local input.

Name: Choose a name that describes the mix of input connectors.

MixerMode Auto: The microphone with the strongest speaker is active and the others are strongly attenuated.

MixerMode Fixed: The input connector signals are mixed together with equal gains.

MixerMode GainShared: The microphones are given a normalized gain factor relative to the strongest speaker before being mixed together.

AGC: Automatic Gain Control.

Mute: Mutes the mix of input connectors.

Channels: Set channels to 1 to mix the input connectors into a mono signal. To mix the input connectors into a stereo signal, set channels to 2.

Requires user role: ADMIN

Parameters:

InputId: <0..65534>
Name: <S: 0, 255>
MixerMode: <Auto/Fixed/GainShared>
AGC: <On/Off>
Mute: <On/Off>
Channels: <1..2>

Example:

```
xCommand Audio LocalInput Add InputId: 2
```

xCommand Audio LocalInput AddConnector

Attach an input connector to the local input given by the input ID. A connector is defined by its type and ID.

This command is mainly intended for internal use by the TC console. TC console automatically generates a start-up script that makes the command persistent over a reboot. To use this command outside of TC console, you have to include it in a start-up script in order to make the setting persistent over a reboot. For more information on star-up scripts and TC console see the administrator guide for your codec.

InputId: A unique identifier for the local input.

ConnectorType: Select the connector type.

ConnectorId: Select a connector.

Requires user role: ADMIN

Parameters:

    InputId(r): <0..65534>
    ConnectorType(r): <HDMI/Line/Microphone>
    ConnectorId(r): <1..8>

Example:

```
xCommand Audio LocalInput AddConnector InputId: 3 ConnectorType: Line
ConnectorId: 1
```

xCommand Audio LocalInput Remove

Remove the local input given by the input ID.

This command is mainly intended for internal use by the TC console. TC console automatically generates a start-up script that makes the command persistent over a reboot. To use this command outside of TC console, you have to include it in a start-up script in order to make the setting persistent over a reboot. For more information on star-up scripts and TC console see the administrator guide for your codec.

InputId: A unique identifier for the local input.

Requires user role: ADMIN

Parameters:

    InputId(r): <0..65534>

Example:

```
xCommand Audio LocalInput Remove InputID: 2
```

xCommand Audio LocalInput RemoveConnector

Detach an input connector from the local input given by the input ID. A connector is defined by its type and ID.

This command is mainly intended for internal use by the TC console. TC console automatically generates a start-up script that makes the command persistent over a reboot. To use this command outside of TC console, you have to include it in a start-up script in order to make the setting persistent over a reboot. For more information on star-up scripts and TC console see the administrator guide for your codec.

InputId: A unique identifier for the local input.

ConnectorType: Select the connector type.

ConnectorId: Select the connector.

Requires user role: ADMIN

Parameters:

    InputId(r): <0..65534>
    ConnectorType(r): <HDMI/Line/Microphone>
    ConnectorId(r): <1..8>

Example:

```
xCommand Audio LocalInput RemoveConnector InputId: 3 ConnectorType: Line
ConnectorId: 1
```

xCommand Audio LocalInput Update

Update the settings of the local input given by the input ID.

This command is mainly intended for internal use by the TC console. TC console automatically generates a start-up script that makes the command persistent over a reboot. To use this command outside of TC console, you have to include it in a start-up script in order to make the setting persistent over a reboot. For more information on star-up scripts and TC console see the administrator guide for your codec.

InputId: A unique identifier for the local input.

Name: Choose a name that describes the mix of input connectors.

MixerMode Auto: The microphone with the strongest speaker is active and the others are strongly attenuated.

MixerMode Fixed: The input connector signals are mixed together with equal gains.

MixerMode GainShared: The microphones are given a normalized gain factor relative to the strongest speaker before being mixed together.

AGC: Automatic Gain Control.

Mute: Mutes the mix of input connectors.

Channels: Set channels to 1 to mix the input connectors into a mono signal. To mix the input connectors into a stereo signal, set channels to 2.

Requires user role: ADMIN

Parameters:

    InputId(r): <0..65534>
    Name(r): <S: 0, 255>
    MixerMode(r): <Auto/Fixed/GainShared>
    AGC(r): <On/Off>
    Mute(r): <On/Off>
    Channels(r): <1..2>

Example:

    xCommand Audio LocalInput Update InputId: 2 Name: "Microphone" MixerMode:
    GainShared AGC: Off Mute: Off Channels: 1

xCommand Audio LocalOutput Add

Create a local output and generate the local output id. A local output is a mix of local input and remote input signals. All connectors attached to the local output receive the same signal.

This command is mainly intended for internal use by the TC console. TC console automatically generates a start-up script that makes the command persistent over a reboot. To use this command outside of TC console, you have to include it in a start-up script in order to make the setting persistent over a reboot. For more information on star-up scripts and TC console see the administrator guide for your codec.

OutputId: A unique identifier for the local output.

Name: Choose a name that describes the local output.

Loudspeaker: If one or more of the output connectors are connected to a loudspeaker, this signal should be a reference signal to the echo canceler. Hence set loudspeaker to On. NOTE: When microphone reinforcement is disabled there should only be one loudspeaker local output.

Channels: Set channels to 1 to mix the local and remote inputs into a mono signal. To mix the inputs into a stereo signal, set channels to 2.

VolumeControlled: Choose whether this output's volume is controlled by global audio control (Off), or if it is at the level defined with xCommand Audio Volume Set (On). If you wish to set VolumeControlled to Off, you must also set the parameter Loudspeaker to Off. Added in TC7.3.3.

Requires user role: ADMIN

Parameters:

    OutputId: <0..65534>
    Name: <S: 0, 255>
    Loudspeaker: <On/Off>
    Channels: <1..2>
    VolumeControlled: <On/Off>

Example:

    xCommand Audio LocalOutput Add OutputId: 47

xCommand Audio LocalOutput AddConnector

Attach an output connector to the local output given by the output ID. A connector is defined by its type and ID.

This command is mainly intended for internal use by the TC console. TC console automatically generates a start-up script that makes the command persistent over a reboot. To use this command outside of TC console, you have to include it in a start-up script in order to make the setting persistent over a reboot. For more information on star-up scripts and TC console see the administrator guide for your codec.

OutputId: A unique identifier for the local output.

ConnectorType: Select the connector type.

ConnectorId: Select a connector.

Requires user role: ADMIN

Parameters:

    OutputId(r): <0..65534>

    ConnectorType(r): <HDMI/Line>

    ConnectorId(r): <1..8>

Example:

```
xCommand Audio LocalOutput AddConnector OutputId:5 ConnectorType: Line
ConnectorId:1
```

xCommand Audio LocalOutput ConnectInput

Connect a local or remote input to a local output by giving their IDs as parameters.

This command is mainly intended for internal use by the TC console. TC console automatically generates a start-up script that makes the command persistent over a reboot. To use this command outside of TC console, you have to include it in a start-up script in order to make the setting persistent over a reboot. For more information on star-up scripts and TC console see the administrator guide for your codec.

OutputId: A unique identifier for the local output.

InputId: A unique identifier for the local input.

InputGain: Set a gain on the input signal in the range from –54dB to 15dB. The value –54dB equals Off.

Requires user role: ADMIN

Parameters:

    OutputId(r): <0..65534>

    InputId(r): <0..65534>

    InputGain: <-54..15>

Example:

```
xCommand Audio LocalOutput ConnectInput OutputId:6 InputId:3
```

xCommand Audio LocalOutput DisconnectInput

Disconnect a local or remote input from a local output.

This command is mainly intended for internal use by the TC console. TC console automatically generates a start-up script that makes the command persistent over a reboot. To use this command outside of TC console, you have to include it in a start-up script in order to make the setting persistent over a reboot. For more information on star-up scripts and TC console see the administrator guide for your codec.

OutputId: A unique identifier for the local output.

InputId: A unique identifier for the local or remote input.

Requires user role: ADMIN

Parameters:

    OutputId(r): <0..65534>

    InputId(r): <0..65534>

Example:

```
xCommand Audio LocalOutput DisconnectInput OutputId:6 InputId:3
```

xCommand Audio LocalOutput Remove

Remove the local output given by the output ID.

This command is mainly intended for internal use by the TC console. TC console automatically generates a start-up script that makes the command persistent over a reboot. To use this command outside of TC console, you have to include it in a start-up script in order to make the setting persistent over a reboot. For more information on star-up scripts and TC console see the administrator guide for your codec.

OutputId: A unique identifier for the local output.

Requires user role: ADMIN

Parameters:

    OutputId(r): <0..65534>

Example:

```
xCommand Audio LocalOutput Remove OutputId: 6
```

xCommand Audio LocalOutput RemoveConnector

Detach an output connector from the local output given by the output ID. A connector is defined by its type and ID.

This command is mainly intended for internal use by the TC console. TC console automatically generates a start-up script that makes the command persistent over a reboot. To use this command outside of TC console, you have to include it in a start-up script in order to make the setting persistent over a reboot. For more information on star-up scripts and TC console see the administrator guide for your codec.

OutputId: A unique identifier for the local output.

ConnectorType: Select the connector type.

ConnectorId: Select the connector.

Requires user role: ADMIN

Parameters:

OutputId(r): <0..65534>
ConnectorType(r): <HDMI/Line>
ConnectorId(r): <1..8>

Example:

```
xCommand Audio LocalOutput RemoveConnector OutputId:5 ConnectorType: Line
ConnectorId:1
```

xCommand Audio LocalOutput Update

Update the settings of the local output given by the output ID.

This command is mainly intended for internal use by the TC console. TC console automatically generates a start-up script that makes the command persistent over a reboot. To use this command outside of TC console, you have to include it in a start-up script in order to make the setting persistent over a reboot. For more information on star-up scripts and TC console see the administrator guide for your codec.

OutputId: A unique identifier for the local output.

Name: Choose a name that describes the local output.

Loudspeaker: If one or more of the output connectors are connected to a loudspeaker, this signal should be a reference signal to the echo canceler. Hence set loudspeaker to On. NOTE: When microphone reinforcement is disabled there should only be one loudspeaker local output.

Channels: Set channels to 1 to mix the local and remote inputs into a mono signal. To mix the inputs into a stereo signal, set channels to 2.

VolumeControlled: Choose whether this output's volume is controlled by global audio control (Off), or if it is at the level defined with xCommand Audio Volume Set (On). If you wish to set VolumeControlled to Off, you must also set the parameter Loudspeaker to Off. Added in TC7.3.3.

Requires user role: ADMIN

Parameters:

OutputId(r): <0..65534>
Name(r): <S: 0, 255>
Loudspeaker(r): <On/Off>
Channels(r): <1..2>
VolumeControlled: <On/Off>

Example:

```
xCommand Audio LocalOutput Update OutputId: 5 Name: "Loudspeaker"
Loudspeaker: On Channels: 2
```

xCommand Audio LocalOutput UpdateInputGain

Update the gain of a local or remote input connected to a local output. The gain on the input signal is in the range from –54dB to 15dB. The value –54dB equals Off.

This command is mainly intended for internal use by the TC console. TC console automatically generates a start-up script that makes the command persistent over a reboot. To use this command outside of TC console, you have to include it in a start-up script in order to make the setting persistent over a reboot. For more information on star-up scripts and TC console see the administrator guide for your codec.

OutputId: A unique identifier for the local output.

InputId: A unique identifier for the local input.

InputGain: Set a gain on the input signal in the range from –54dB to 15dB. The value –54dB equals Off.

Requires user role: ADMIN

Parameters:

OutputId(r): <0..65534>
InputId(r): <0..65534>
InputGain: <-54..15>

Example:

xCommand Audio LocalOutput UpdateInputGain OutputId:6 InputId:3

xCommand Audio Microphones Mute

Mute all microphones.

Requires user role: USER

Example:

xCommand Audio Microphones Mute

xCommand Audio Microphones Unmute

Unmute microphones.

Requires user role: USER

Example:

xCommand Audio Microphones Unmute

xCommand Audio RemoteOutput ConnectInput

Connect a local or remote input to a remote output with their IDs as parameters. When a call is made a remote input and remote output pair is created.

This command is mainly intended for internal use by the TC console. TC console automatically generates a start-up script that makes the command persistent over a reboot. To use this command outside of TC console, you have to include it in a start-up script in order to make the setting persistent over a reboot. For more information on star-up scripts and TC console see the administrator guide for your codec.

OutputId: A unique identifier for the local output.

InputId: A unique identifier for the local or remote input.

InputGain: Set a gain on the input signal in the range from –54dB to 15dB. The value –54dB equals Off.

Requires user role: ADMIN

Parameters:

OutputId(r): <0..65534>
InputId(r): <0..65534>
InputGain: <-54..15>

Example:

xCommand Audio RemoteOutput ConnectInput OutputId:10 InputId:8

xCommand Audio RemoteOutput DisconnectInput

Disconnect a local or remote input from a remote output with their IDs as parameters.

This command is mainly intended for internal use by the TC console. TC console automatically generates a start-up script that makes the command persistent over a reboot. To use this command outside of TC console, you have to include it in a start-up script in order to make the setting persistent over a reboot. For more information on star-up scripts and TC console see the administrator guide for your codec.

OutputId: A unique identifier for the local output.

InputId: A unique identifier for the local or remote input.

Requires user role: ADMIN

Parameters:

OutputId(r): <0..65534>
InputId(r): <0..65534>

Example:

xCommand Audio RemoteOutput DisconnectInput OutputId:10 InputId:8

xCommand Audio RemoteOutput UpdateInputGain

Update the gain of a local or remote input connected to a remote output.

This command is mainly intended for internal use by the TC console. TC console automatically generates a start-up script that makes the command persistent over a reboot. To use this command outside of TC console, you have to include it in a start-up script in order to make the setting persistent over a reboot. For more information on star-up scripts and TC console see the administrator guide for your codec.

OutputId: A unique identifier for the local output.

InputId: A unique identifier for the local or remote input.

InputGain: Set a gain on the input signal in the range from -54dB to 15dB. The value -54dB equals Off.

Requires user role: ADMIN

Parameters:

OutputId(r): <0..65534>
InputId(r): <0..65534>
InputGain: <-54..15>

Example:

xCommand Audio RemoteOutput UpdateInputGain OutputId:6 InputId:3

xCommand Audio Setup Clear

Remove all local inputs and local outputs.

Requires user role: ADMIN

Example:

xCommand Audio Setup Clear

xCommand Audio Sound Play

Play a specified audio sound.

Requires user role: USER

Parameters:

Sound(r): <Bump/Busy/CallDisconnect/CallInitiate/CallWaiting/Dial/KeyInput/KeyTone/Nav/NavBack/Notification/OK/PresentationConnect/Ringing/SpecialInfo/TelephoneCall/VideoCall/Volume>
Loop: <On/Off>

Example:

xCommand Audio Sound Play Sound: Ringing

xCommand Audio Sound Stop

Stop playing audio sound.

Requires user role: USER

Example:

xCommand Audio Sound Stop

xCommand Audio SoundsAndAlerts Ringtone List

Lists all available ringtones that can be configured using xConfiguration Audio SoundsAndAlerts RingTone.

Requires user role: USER

Example:

xCommand Audio SoundsAndAlerts Ringtone List
*r AudioRingtoneListResult Ringtone 1 Id: "Sunrise"
*r AudioRingtoneListResult Ringtone 2 Id: "Mischief"
*r AudioRingtoneListResult Ringtone 3 Id: "Ripples"
*r AudioRingtoneListResult Ringtone 4 Id: "Reflections"
*r AudioRingtoneListResult Ringtone 5 Id: "Vibes"
*r AudioRingtoneListResult Ringtone 6 Id: "Delight"
*r AudioRingtoneListResult Ringtone 7 Id: "Evolve"
*r AudioRingtoneListResult Ringtone 8 Id: "Playful"
*r AudioRingtoneListResult Ringtone 9 Id: "Ascent"
*r AudioRingtoneListResult Ringtone 10 Id: "Calculation"
*r AudioRingtoneListResult Ringtone 11 Id: "Mellow"
*r AudioRingtoneListResult Ringtone 12 Id: "Ringer"

xCommand Audio SoundsAndAlerts Ringtone Play

Play one of the available ringtones. To get a list of the available ringtones use the command xCommand Audio SoundsAndAlerts Ringtone List.

Requires user role: USER

Parameters:

RingTone(r): <S: 1, 100>

Example:

xCommand Audio SoundsAndAlerts Ringtone Play RingTone: Sunrise

xCommand Audio Volume Decrease

Decrease the volume on the endpoint.

Requires user role: USER

Parameters:

Steps: <1..10>

Example:

```
xCommand Audio Volume Decrease Steps:3
```

xCommand Audio Volume Increase

Increase the volume on the endpoint.

Requires user role: USER

Parameters:

Steps: <1..10>

Example:

```
xCommand Audio Volume Increase Steps:3
```

xCommand Audio Volume Mute

Mute the volume on the endpoint.

Requires user role: USER

Example:

```
xCommand Audio Volume Mute
```

xCommand Audio Volume Set

Set the volume on the endpoint to a specified level.

Requires user role: USER

Parameters:

Level(r): <1..100>

Example:

```
xCommand Audio Volume Set Level:30
```

xCommand Audio Volume SetToDefault

Set the current volume level as the default for the endpoint.

Requires user role: USER

Example:

```
xCommand Audio Volume SetToDefault
```

xCommand Audio Volume UnMute

Unmute the volume on the endpoint.

Requires user role: USER

Example:

```
xCommand Audio Volume SetToDefault
```

xCommand Audio VUMeter Start

Start collecting VU meter information for the connector identified by the ConnectorId and ConnectorType parameters. Together these parameters identify the connector.

ConnectorId: The connector identifier is printed on the codec connector panel.

ConnectorType: Identifies whether it is an HDMI input connector, an audio line input connector or a microphone connector.

Requires user role: USER

Parameters:

ConnectorType(r): <HDMI/Line/Microphone>
ConnectorId(r): <1..8>

Example:

```
xCommand Audio VUMeter Start ConnectorType: Microphone ConnectorId: 1
```

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

96

xCommand Audio VUMeter Stop

Stop collecting VU meter information for the connector identified by the ConnectorId and ConnectorType parameters. Together these parameters uniquely identify the connector.

ConnectorId: The connector identifier is printed on the codec connector panel.

ConnectorType: Identifies whether it is an HDMI input connector, an audio line input connector or a microphone connector.

Requires user role: USER

Parameters:

ConnectorType(r): <HDMI/Line/Microphone>
ConnectorId(r): <1..8>

Example:

```
xCommand Audio VUMeter Stop ConnectorType: Microphone ConnectorId: 1
```

xCommand Audio VUMeter StopAll

Stop collecting VU meter information for all connectors.

Requires user role: USER

Example:

```
xCommand Audio VUMeter StopAll
```

Bookings commands

xCommand Bookings Clear

Clear the current stored list of bookings.

Requires user role: USER

Example:

```
xCommand Bookings Clear
```

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


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
```

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.
```

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. CallType parameter is not in use.

Requires user role: USER

Parameters:

CallId: <0..65534>
CallType: <Audio/Video> Not in use.

Example:

```
xCommand Call Accept CallId:19
```

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
```

xCommand Call DisconnectAll

Disconnect all active calls.

Requires user role: USER

Example:

```
xCommand Call DisconnectAll
```

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: <Conference/Transfer/Other>

Example:

xCommand Call Hold CallId:19

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

xCommand Call Join

Join all existing calls, active and on hold.

Requires user role: USER

Parameters:

CallId(r): <0..65534>

Example:

xCommand Call Join CallId: 34

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

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

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

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

100

Copyright © 2015 Cisco Systems, Inc. All rights reserved.

CallHistory commands

xCommand CallHistory AcknowledgeAllMissedCalls

Turns off the missed calls indicator on the touch controller for all missed calls.

Requires user role: USER

Example:

```
xCommand CallHistory AcknowledgeAllMissedCalls
```

xCommand CallHistory AcknowledgeMissedCall

Turns off the missed calls indicator on the touch controller for the specified call.

CallHistoryId: CallHistoryId for the call in question.

AcknowledgeConsecutiveDuplicates: Includes all surrounding calls with duplicate information.

Requires user role: USER

Parameters:

CallHistoryId(r): <1..2147483647>

AcknowledgeConsecutiveDuplicates: <False/True>

Example:

```
xCommand CallHistory AcknowledgeMissedCalls CallHistoryId: 5
```

xCommand CallHistory DeleteAll

Deletes all information on previous calls.

Filter: You can filter which calls to delete, either all, missed, placed or received calls.

Requires user role: USER

Parameters:

Filter: <All/Missed/Placed/Received>

Example:

```
xCommand CallHistory DeleteAll Filter: Missed
```

xCommand CallHistory DeleteEntry

Deletes all information on the specified call.

CallHistoryId: CallHistoryId for the call in question.

AcknowledgeConsecutiveDuplicates: Includes all surrounding calls with duplicate information.

Requires user role: USER

Parameters:

CallHistoryId(r): <1..2147483647>

AcknowledgeConsecutiveDuplicates: <False/True>

Example:

```
xCommand CallHistory DeleteEntry CallHistoryId: 5
```

xCommand CallHistory Get

Retrieve all information on previous calls made on the video system.

Filter: You can filter which calls to retrieve.

Offset: Sets the call from which to start.

Limit: Defines the amount of calls in the output.

DetailLevel: Sets the level of detail for the information on these calls.

SearchString: Allows you to set the command to apply to a specified display name or call back number.

CallHistoryId: CallHistoryId for the call in question.

Requires user role: USER

Parameters:

Filter: <All/Missed/AnsweredElsewhere/Forwarded/Placed/NoAnswer/Received/Rejected/UnacknowledgedMissed>

Offset: <0..65534>

Limit: <0..65534>

DetailLevel: <Basic/Full>

SearchString: <S: 0, 255>

CallHistoryId: <0..65534>

Example:

```
xCommand CallHistory Get Filter: All Offset: 3 DetailLevel: Basic

*r CallHistoryGetResult Entry 0 CallHistoryId: 2

*r CallHistoryGetResult Entry 0 CallbackNumber: "sip:room1@company.com"

*r CallHistoryGetResult Entry 0 DisplayName: "display.name"

*r CallHistoryGetResult Entry 0 StartTime: "2013-04-09T10:56:36"

*r CallHistoryGetResult Entry 0 DaysAgo: 182

*r CallHistoryGetResult Entry 0 OccurrenceType: Received

*r CallHistoryGetResult Entry 0 IsAcknowledged: Acknowledged

*r CallHistoryGetResult Entry 1 CallHistoryId: 1

*r CallHistoryGetResult Entry 1 CallbackNumber: "sip:name@company.com"

*r CallHistoryGetResult Entry 1 DisplayName: "name"

*r CallHistoryGetResult Entry 1 StartTime: "2013-04-09T10:53:53"

*r CallHistoryGetResult Entry 1 DaysAgo: 182

*r CallHistoryGetResult Entry 1 OccurrenceType: Received

*r CallHistoryGetResult Entry 1 IsAcknowledged: Acknowledged

*r CallHistoryGetResult ResultInfo Offset: 3

*r CallHistoryGetResult ResultInfo Limit: 65534
```

xCommand CallHistory Recents

Retrieve aggregated information on previous calls made on the video system.

Filter: You can filter which calls to retrieve.

Offset: Sets the call from which to start.

Limit: Defines the amount of calls in the output.

DetailLevel: Sets the level of detail for the information on these calls.

SearchString: Allows you to set the command to apply to a specified display name or call back number.

CallHistoryId: CallHistoryId for the call in question.

Order: Define the order in which the previous calls are presented.

Requires user role: USER

Parameters:

Filter: <All/Missed/AnsweredElsewhere/Forwarded/Placed/NoAnswer/Received/Rejected/UnacknowledgedMissed>

Offset: <0..65534>

Limit: <0..65534>

DetailLevel: <Basic/Full>

SearchString: <S: 0, 255>

CallHistoryId: <0..65534>

Order: <OccurrenceTime/OccurrenceFrequency>

Example:

```
xCommand CallHistory Recents Filter: Missed Offset: 6 DetailLevel: Full
```

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 Duration: 30

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..7>

Example:

xCommand Camera PanTiltReset CameraId:1

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..7>

PresetId(r): <1..15>

Example:

xCommand Camera PositionActivateFromPreset CameraId:1 PresetId:1

xCommand Camera PositionReset

Reset the camera position the the default position.

Requires user role: USER

Parameters:

CameraId(r): <1..7>

Example:

xCommand Camera PositionReset CameraId:1

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..7>

Pan: <-65535..65535>

Tilt: <-65535..65535>

Zoom: <0..65535>

Focus: <0..65535>

Example:

```
xCommand Camera PositionSet CameraId:1 Pan:200 Tilt:300
```

xCommand Camera Preset Activate

Activate one of the stored camera presets.

Note that 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
```

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 that 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: USER

Parameters:

PresetId(r): <1..35>

ListPosition: <1..35>

Name: <S: 0, 255>

Example:

```
xCommand Camera Preset Edit PresetId: 1 ListPosition: 1 Name: ""
```

xCommand Camera Preset List

List information about available camera presets.

Note that 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: Filter on specified preset.

Requires user role: USER

Parameters:

PresetId: <1..35>

Example:

```
xCommand Camera Preset List PresetId: 1
```

xCommand Camera Preset Remove

Remove a camera preset.

Note that 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

xCommand Camera Preset Snapshot Get

Retrieve a stored snapshot image to illustrate a camera preset.
PresetId: Preset id for the snapshot.

Requires user role: USER

Parameters:

PresetId(r): <1..35>

Example:

xCommand Camera Preset Snapshot Get PresetId:1

xCommand Camera Preset Snapshot Remove

Remove a stored snapshot.
PresetId: Preset id for the snapshot.

Requires user role: ADMIN

Parameters:

PresetId(r): <1..35>

Example:

xCommand Camera Preset Snapshot Remove PresetId:1

xCommand Camera Preset Snapshot Store

Store a new snapshot. This command can only be used if snapshots are enabled with xConfiguration Video AllowWebSnapshots.

PresetId: Preset id for the snapshot.

Requires user role: ADMIN

Parameters:

PresetId(r): <1..35>

Example:

xCommand Camera Preset Snapshot Store PresetId:1

xCommand Camera Preset Store

Store the current position (pan and tilt), zoom and focus of the chosen camera. The camera is identified by the CameraId parameter.

Note that 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. The individual camera presets are not available for far end control.

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.

TakeSnapshot: Allow or disallow snapshot of the preview.

Requires user role: USER

Parameters:

PresetId: <1..35>
CameraId (r): <1..7>
ListPosition: <1..35>
Name: <S: 0, 255>
TakeSnapshot; <False/True>

Example:

xCommand Camera Preset Store PresetId: 1 CameraId: 1 ListPosition: 1
Name: ""

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..7>

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
xCommand Camera Ramp CameraId:1 Pan:stop
```

xCommand Camera ReconfigureCameraChain

Reinitialize the daisy chain of cameras and updates the CameraId parameter. The CameraId parameter holds information of which camera is sitting in what position in the camera chain.

Requires user role: USER

Example:

```
xCommand Camera ReconfigureCameraChain
```

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..7>

Example:

```
xCommand Camera TriggerAutofocus CameraId:1
```

Cameras commands

xCommand Cameras SpeakerTrack Activate

Activate SpeakerTrack mode. Requires that xConfiguration Cameras SpeakerTrack Mode is set to Auto (default).

This setting applies only when using a Cisco TelePresence SpeakerTrack 60 camera.

The SpeakerTrack 60 camera assembly consists of two cameras and uses an audio tracking technique that finds and captures a close-up of the active speaker. When a change of speaker is detected, the system can switch automatically between the two cameras to always show the best camera view.

Requires user role: USER

Example:

```
xCommand Cameras SpeakerTrack Activate
```

xCommand Cameras SpeakerTrack Deactivate

Deactivate SpeakerTrack mode.

This setting applies only when using a Cisco TelePresence SpeakerTrack 60 camera.

The SpeakerTrack 60 camera assembly consists of two cameras and uses an audio tracking technique that finds and captures a close-up of the active speaker. When a change of speaker is detected, the system can switch automatically between the two cameras to always show the best camera view.

Requires user role: USER

Example:

```
xCommand Cameras SpeakerTrack Deactivate
```

xCommand Cameras SpeakerTrack Whiteboard ActivatePosition

Moves the specified camera in a SpeakerTrack 60 to the position stored with xCommand Cameras SpeakerTrack Whiteboard StorePosition.

Camera: Select either right or left camera.

WhiteboardId: The id set for the camera position used for the snap-to-whiteboard feature. Currently only id 1 can be used.

Requires user role: ADMIN

Parameters:

Camera (r): <Left/Right>

WhiteboardId: <0-65534>

Example:

```
xCommand Cameras SpeakerTrack Whiteboard ActivatePosition Camera:Left
WhiteboardId:123
```

xCommand Cameras SpeakerTrack Whiteboard AlignPosition

Align the position of the other camera on a SpeakerTrack 60 to the first cameras stored position. You must also set the distance to a whiteboard.

Camera: Select either right or left camera.

Distance: Set distance to whiteboard in millimeters. Use an accurate measurement for the feature to work seamlessly.

Requires user role: ADMIN

Parameters:

Camera (r): <Left/Right>

Distance (r): <3000-15000>

Example:

```
xCommand Cameras SpeakerTrack Whiteboard AlignPosition Camera:Left
Distance:8000
```

xCommand Cameras SpeakerTrack Whiteboard SetDistance

Set the cameras distance to the whiteboard. This information is needed by the SpeakerTrack 60 to frame the whiteboard automatically.

Distance: Set distance to whiteboard in centimeters. Use a distance between 0-2000 cm. Use an accurate measurement for the feature to work seamlessly.

WhiteboardId: The id set for the camera position used for the snap-to-whiteboard feature. Currently only id 1 can be used.

Requires user role: ADMIN

Parameters:

Distance (r): <0-65534>

WhiteboardId: <0-65534>

Example:

```
xCommand Cameras SpeakerTrack Whiteboard SetDistance Distance:1000
WhiteboardId:123
```

xCommand Cameras SpeakerTrack Whiteboard StorePosition

Store the position of one of the cameras on a SpeakerTrack 60 as the Snap to Whiteboard position. Frame the image so that there is room around the whiteboard for the speaker. To use the Snap to Whiteboard feature it must be enabled with xConfiguration Cameras SpeakerTrack Whiteboard Mode and tracking must be enabled with xConfiguration Cameras SpeakerTrack Mode.

Note: You must store a snap-to-whiteboard position for both cameras.

Camera: Select the camera which position you wish to store.

WhiteboardId: You can give the position an id number. Currently the only id that can be used is 1.

Requires user role: ADMIN

Parameters:

Camera (r): <Left/Right>

WhiteboardId: <0-65534>

Example:

```
xCommand Cameras SpeakerTrack Whiteboard StorePosition Camera:Left
WhiteboardId:123
```

Conference commands

xCommand Conference ActiveSpeaker Reset

Resets ActiveSpeaker. Normal voice switching is switched back on.

Requires user role: USER

Example:

```
xCommand Conference ActiveSpeaker Reset
```

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


Example 2
xCommand Conference ActiveSpeaker Set Target: remote CallID: 3
```

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 Timeout: 120
```

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
```

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 Auto, 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.

DisplayName: The display name of the remote participant.

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: <1..999999999>

DisplayName: <S: 0, 255>

Example:

```
xCommand Dial Number:543210 Protocol:h323
```

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
```

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
```

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
```

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
```

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
```

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: USER

Parameters:

CallId: <0..65534>

PresetId(r): <0..15>

Example:

```
xCommand FarEndControl Preset Store CallId:3 PresetId:1
```

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

111

Copyright © 2015 Cisco Systems, Inc. All rights reserved.

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
```

GPIO commands

xCommand GPIO ManualState Set

Set the status of a GPIO pin that is configured to OutputManualState mode (ref. xConfiguration GPIO Pin [1..4] Mode).

Requires user role: USER

Parameters:

Pin1: <High/Low>

Pin2: <High/Low>

Pin3: <High/Low>

Pin4: <High/Low>

Example:

```
xCommand GPIO ManualState Set Pin1: High
```

HttpFeedback commands

xCommand HttpFeedback Deregister
Deregister XML feedback over HTTP(S).

Requires user role: USER

Parameters:
FeedbackSlot(r): <1..4>

Example:
```
xCommand HttpFeedback Deregister FeedbackSlot:1
```

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
```

Key commands

xCommand Key Click
Not applicable for this product.

xCommand Key Press
Not applicable for this product.

xCommand Key Release
Not applicable for this product.

Logging commands

xCommand Logging ExtendedLogging Start

Turns on the extended logging, which stores more information than regular logging in the log files. You can determine the duration in seconds. Note that extended logging uses more of your video system's resources, and should only be used when troubleshooting an issue.

Requires user role: USER

Parameters:

Duration: <1..600>

Example:

```
xCommand Logging ExtendedLogging Start Duration: 20
```

xCommand Logging ExtendedLogging Stop

Stops the logging started with xCommand SystemUnit Diagnostics ExtendedLogging Start.

Requires user role: USER

Example:

```
xCommand Logging ExtendedLogging Stop
```

Message commands

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
```

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
```

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"
```

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: USER

Parameters:

CallId: <0..65534>

Type(r): <S: 0, 255>

Text(r): <S: 1, 1450>

LocalEcho: <On/Off>

Example:

```
xCommand Message FarendMessage CallId: 3 Type: 3 Text: Hello
```

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 FeedbackId: 24
```

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"
```

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 xFeed-back 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: USER

Parameters:

FeedbackId: <S: 0, 255>

OptionId(r): <1..5>

Example:

```
xCommand Message Prompt Response OptionId: 1
```

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
```

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
```

Peripherals commands

xCommand Peripherals Connect

Register peripherals that are connected to the codec, such as control systems and touch panels. The registered peripherals are displayed on the web interface under Configuration > Peripherals.

This command should be used when the peripheral connects to the codec for the first time or when the software version on the peripheral has changed. The list of connected devices is available with the command xStatus Peripherals ConnectedDevice [n] Status.

Note: Does not apply for the Cisco PrecisionHD cameras.

Requires user role: USER

Parameters:

HardwareInfo: <S: 0, 100>
ID(r): <S: 1, 100>
Name: <S: 0, 100>
NetworkAddress: <S: 0, 100>
SerialNumber: <S: 0, 100>
SoftwareInfo: <S: 0, 100>
Type(r): <Byod, ControlSystem, Other, TouchPanel>

Example:

```
xCommand Peripherals Connect ID:"00:50:60:d8:7c:e6" Type: Other
```

xCommand Peripherals HeartBeat

When a peripheral is registered as a connected device, you can set it to send a heartbeat to the codec to let the codec know that it is still connected.

This will keep the device on the xStatus Peripherals ConnectedDevice list. If the peripheral is not set to send a heartbeat, the device will disappear from the list after a while.

Note: Does not apply for ISDN Link or cameras.

Requires user role: USER

Parameters:

ID(r): <S: 1, 100>
Timeout: <1..65535>

Example:

```
xCommand Peripherals HeartBeat ID: "00:50:60:d8:7c:e6"
```

xCommand Peripherals List

Lists all currently and previously connected peripherals.

Note: Does not apply for the Cisco PrecisionHD cameras.

Connected: Limit the search to currently connected devices.

Type: Limit the search by device type.

Requires user role: ADMIN

Parameters:

Connected: <False, True>
Type: <All, BluetoothHeadset, ControlSystem, ISDNLink, Other, TouchPanel>

Example:

```
xCommand Peripherals List Connected: True Type: Other
```

xCommand Peripherals ManualUpgrade

Manually upgrade software on a Precision 60 or a SpeakerTrack 60 camera.

Password: Enter a password, if one is needed to upload the software package.

Product: Define the device that will be upgraded.

URL: Define the URL from which to upload the software package.

Username: Enter a username, if one is needed to upload the software package.

Requires user role: ADMIN

Parameters:

Password: <S: 0, 64>
Product(r): <Precision 60 Camera, SpeakerTrack 60>
URL(r): <S: 0, 1024>
Username: <S: 0, 64>

Example:

```
xCommand Peripherals ManualUpgrade Product: Precision 60 Camera URL:
"http://softwareupload.company.com"
```

xCommand Peripherals Pairing DeviceDiscovery Start

Start device discovery to detect ISDN Links in the same network.

AutoPairing: You can select to automatically pair the detected device to the endpoint.

DeviceType: Only look for ISDN Link.

Timeout: Set a maximum time for the search from 3 to 60 seconds.

Requires user role: ADMIN

Parameters:

AutoPairing: <On, Off>

DeviceType: <ISDNLink>

Timeout: <3..60>

Example:

```
xCommand Peripherals Pairing DeviceDiscovery Start
```

xCommand Peripherals Pairing Pair

Pair an ISDN Link to an endpoint.

MacAddress: Enter the MAC address for the ISDN Link you wish to pair to the endpoint.

Requires user role: ADMIN

Parameters:

MacAddress(r): <S: 1, 1450>

Example:

```
xCommand Peripherals Pairing Pair MacAddress: 1.2.3.4
```

xCommand Peripherals Pairing Unpair

Unpair endpoint from an ISDN Link, when the two have contact.

MacAddress: Enter the MAC address for the ISDN Link you wish to unpair from the endpoint.

Requires user role: ADMIN

Parameters:

MacAddress(r): <S: 1, 100>

Example:

```
xCommand Peripherals Pairing Unpair MacAddress: 1.2.3.4
```

xCommand Peripherals Pairing Purge

Force unpair an endpoint from an ISDN Link when a connection has been lost. Note: You must also unpair the ISDN Link to be able to pair it to another endpoint.

Requires user role: ADMIN

Parameters:

ID(r): <S: 1, 100>

Example:

```
xCommand Peripherals Purge ID: 1.2.3.4
```

Phonebook commands

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 and on the web interface.

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 the Auto, SIP, H323 or H320 protocol.

CallRate: Set a call rate.

CallType: Select a call type (audio or video).

Device: Select the device type.

Tag: Tag the contact as a Favorite, or untag an already tagged contact.

Requires user role: USER

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>

    Tag: <Untagged/Favorite>

Example:

```
xCommand Phonebook Contact Add Name: "John Doe" Number:12345
```

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: USER

Parameters:

    ContactId(r): <S: 0, 255>

Example:

```
xCommand Phonebook Contact Delete ContactId:localContactId-1
```

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.

Tag: Tag the contact as a Favorite, or untag an already tagged contact.

Requires user role: USER

Parameters:

    ContactId(r): <S: 0, 255>

    Name: <S: 0, 255>

    FolderId: <S: 0, 255>

    ImageURL: <S: 0, 255>

    Title: <S: 0, 255>

    Tag: <Untagged/Favorite>

Example:

```
xCommand Phonebook Contact Modify ContactId:localContactId-1 Name: "John
Doe - office"
```

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 and on the web interface.

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: USER

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
```

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: USER

Parameters:

ContactId(r): <S: 0, 255>

ContactMethodId(r): <S: 0, 255>

Example:

```
xCommand Phonebook ContactMethod Delete ContactId:localContactId-2
ContactMethodId:1
```

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: USER

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"
```

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: USER

Parameters:

  Name(r): <S: 0, 255>

  ParentFolderId: <S: 0, 255>

Example:

```
xCommand Phonebook Folder Add Name: "New York Office"
```

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: USER

Parameters:

  FolderId(r): <S: 0, 255>

Example:

```
xCommand Phonebook Folder Delete FolderId:localGroupId-3
```

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: USER

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"
```

xCommand Phonebook Search

The search command lets you search in both the local and corporate phone books. A search gives a ResultSet.

The total number of folders and contacts (TotalRows) is always included in the result set when searching the local phone book. When searching a corporate phonebook the total number of folders and contacts may not be included. Whether it is included or not depends on the backend corporate phonebook service (e.g. CUCM, VCS, TMS) and its version.

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.

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.

Tag: Limits the search to either contacts that have been tagged as favorite or the untagged contacts.

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: <Any/Folder/Contact>

Tag: <Untagged/Favorite>

Example:

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

122

Copyright © 2015 Cisco Systems, Inc. All rights reserved.

```
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
```

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.

Instance: Select which local presentation you wish to start.

Requires user role: USER

Parameters:

PresentationSource: <1..4>
ConnectorId: <1..5>
SendingMode: <LocalRemote/LocalOnly>
Instance: <New / 1 / 2 / 3 / 4 / 5 / 6>
SendingMode: <1 – 4>

Example:

```
xCommand Presentation Start PresentationSource:2
```

xCommand Presentation Stop

Stop the media stream from the presentation source.

Instance: Select which local presentation you wish to stop.

Requires user role: USER

Parameters:

Instance: <1 / 2 / 3 / 4 / 5 / 6>

Example:

```
xCommand Presentation Stop
```

Preset commands

xCommand Preset Activate

Activate one of the stored local presets.

Note that information about all video input sources, and pan, tilt, zoom and focus values for all cameras are included in the same preset. In contrast, the xCommand Camera Preset commands applies to individual cameras only.

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
```

xCommand Preset Clear

Delete a preset.

Note that information about all video input sources, and pan, tilt, zoom and focus values for all cameras are included in the same preset. In contrast, the xCommand Camera Preset commands applies to individual cameras only.

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
```

xCommand Preset Store

Store the connector selections for all video input sources and the current position (pan and tilt), zoom and focus values for all cameras.

Note that information about all video input sources, and pan, tilt, zoom and focus values for all cameras are included in the same preset. The system may hold 15 such 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: Not applicable. 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: USER

Parameters:

PresetId(r): <1..15>

Type(r): <All/Camera>

Description: <S: 0, 255>

Example:

```
xCommand Preset Store PresetId:3 Type:Camera Description:"Left view"
```

Provisioning commands

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: <S: 4, 10>

Example:

```
xCommand Provisioning CUCM CAPF OperationStart AuthString: 123456
```

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
```

xCommand Provisioning CUCM CTL Show

Shows the content of the installed Certificate Trust List file (CTL), if it exists. Each entry displayed contains the information about one specific certificate. If a certificate has been deleted in CTL, it is marked accordingly in the output.

Requires user role: USER

Example:

```
xCommand Provisioning CUCM CTL Show
```

xCommand Provisioning CUCM ExtensionMobility Login

Login command for the Extension Mobility service. You login to the Extension Mobility service with a user name and pin. If there are multiple profiles available for that user, a separate event will be sent with the available profiles for the user. A new login command with the profile can then be submitted.

Requires user role: USER

Parameters:
   UserId(r): <S: 1, 255>
   Pin(r): <S: 1, 255>
   Profile: <S: 1, 255>

Example:

```
xCommand Provisoning CUCM ExtensionMobility Login UserId: user1 Pin: 1234
```

xCommand Provisioning CUCM ExtensionMobility Logout

This command will log you out of your Extension Mobility profile.

Requires user role: USER

Example:

```
xCommand Provisioning CUCM ExtensionMobility Logout
```

xCommand Provisioning CUCM ITL Show

Shows the content of the installed Identity Trust List file (ITL), if it exists. Each entry displayed contains the information about one specific certificate. If a certificate has been deleted in CTL, it is marked accordingly in the output.

Requires user role: USER

Example:

```
xCommand Provisioning CUCM ITL Show
```

xCommand Provisioning CompleteUpgrade

Starts installing the software upgrade if you wish to install it before it is set to do so.

Requires user role: USER

Example:

```
xCommand Provisioning CompleteUpgrade
```

xCommand Provisioning PostponeUpgrade

Postpones the installing of the software upgrade.

Requires user role: USER

Parameters:

SecondsToPostpone(r): <0..65534>

Example:

```
xCommand Provisioning PostponeUpgrade SecondsToPostpone: 60
```

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
```

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 remote support user is not available.

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"
```

xCommand Security Persistency

Set the following features to persistent or non-persistent mode. In non-persistent mode the information gathered by the specified feature does not persist a reboot of the system. Persistent mode is the default. This command reboots the system.

Configuration: In non-persistent mode, all configurations are set back to default when the system reboots.

CallHistory: In non-persistent mode call history is deleted when the system reboots.

InternalLogging: In non-persistent mode eventlog is deleted when the system reboots.

LocalPhonebook: In non-persistent mode local phone book is deleted when the system reboots.

DHCP: In non-persistent mode all IP related information is deleted when the system reboots.

ConfirmandReboot: Reboots the system.

Requires user role: ADMIN

Parameters:

Configurations(r): <NonPersistent/Persistent>

CallHistory(r): <NonPersistent/Persistent>

InternalLogging(r): <NonPersistent/Persistent>

LocalPhonebook(r): <NonPersistent/Persistent>

DHCP(r): <NonPersistent/Persistent>

ConfirmAndReboot(r): <Yes>

Example:

```
xCommand Security Persistency Configurations: Persistent CallHistory:
Persistent InternalLogging: Persistent LocalPhonebook: Persistent DHCP:
Persistent ConfirmAndReboot: Yes
```

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
```

xCommand Standby Deactivate

Bring the system out of standby mode.

Requires user role: USER

Example:

```
xCommand Standby Deactivate
```

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
```

SystemUnit commands

xCommand SystemUnit AdminPassword Set

Set a password for the user that is currently signed in.

Requires user role: ADMIN

Parameters:

Password(r): <S: 0, 64>

Example:

```
xCommand SystemUnit AdminPassword Set Password:***********
```

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
```

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 List

List configuration profiles that has been stored in the system.

Requires user role: ADMIN

Example:

```
xCommand SystemUnit ConfigurationProfile List
```

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"
```

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"
```

xCommand SystemUnit DateTime Get

Read the time and date from the system.

Requires user role: USER

Example:

```
xCommand SystemUnit DateTime Get
```

128

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
```

xCommand SystemUnit Diagnostics Run

This command runs all self-diagnostics commands on the system.

Requires user role: ADMIN

Parameters:

ResultSet: <Alerts/All/None>

Example:

```
xCommand SystemUnit Diagnostics Run
```

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. Option key(s) 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
```

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
```

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:******************
```

xCommand SystemUnit OptionKey List

List all option keys that have been added to the system. Added in TC7.3.3.

Requires user role: ADMIN

Example:

```
xCommand SystemUnit OptionKey List
```

xCommand SystemUnit OptionKey Remove

Remove an individual Option key. Added in TC7.3.3.

Requires user role: ADMIN

Parameters:

Type(r): <Encryption/MultiSite/PremiumResolution/RemoteMonitoring>

Example:

```
xCommand SystemUnit OptionKey Remove Type: PremiumResolution
```

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

129

Copyright © 2015 Cisco Systems, Inc. All rights reserved.

xCommand SystemUnit OptionKey RemoveAll

Remove all Option keys.

Requires user role: ADMIN

Parameters:

Confirm(r): <Yes>

Example:

```
xCommand SystemUnit OptionKey RemoveAll Confirm: Yes
```

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
```

UserInterface commands

xCommand UserInterface OSD Close

Clear all messages and indicators from the monitor(s).

Requires user role: USER

Parameters:

Element(r): <Menu>

Example:

```
xCommand UserInterface OSD Close Element: Menu
```

xCommand UserInterface ScreenShot Get

For internal use only.

Requires user role: ADMIN

Parameters:

ScreenShotId(r): <S: 0, 50>

Example:

```
xCommand UserInterface ScreenShot Get ScreenShotId:5
```

xCommand UserInterface ScreenShot List

For internal use only.

Requires user role: ADMIN

Example:

```
xCommand UserInterface ScreenShot List
```

xCommand UserInterface ScreenShot Remove

For internal use only.

Requires user role: ADMIN

Parameters:

ScreenShotId(r): <S: 0-50>

Example:

```
xCommand UserInterface ScreenShot Remove ScreenShotId:5
```

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

130

Copyright © 2015 Cisco Systems, Inc. All rights reserved.

xCommand UserInterface ScreenShot Request

For internal use only.

Requires user role: ADMIN

Parameters:

Type(r): <Touchpanel/OSD>

ScreenShotId(r): <S: 0-50>

Example:

```
xCommand UserInterface ScreenShot Request ScreenShotId: 3 Type: OSD
```

xCommand UserInterface ScreenShot Store

For internal use only.

Requires user role: USER

Parameters:

Type(r): <Touchpanel/OSD>

Name: <S: 0, 35>

ScreenShotId(r): <S: 0-50>

Example:

```
xCommand UserInterface ScreenShot Store Type:OSD Name:5 ScreenShotId:5
```

UserManagement commands

xCommand UserManagement RemoteSupportUser Create

Create a remote support user passphrase that Technical Assistance Center (TAC) can use to access the system for troubleshooting. You can set duration for the validity of the passphrase. It expires in 7 days by default.

Requires user role: ADMIN

Parameters:

ExpiryDays: <1..31>

Example:

```
xCommand UserManagement RemoteSupportUser Create

*r UserCreate (status=OK):
    Username: remotesupport
    Expiry: 2014-03-04 13:24:15 UTC
    Passphrase: 1234567890
```

xCommand UserManagement RemoteSupportUser Delete

Delete the remote support user created with the command xCommand UserManagement RemoteSupportUser Create.

Requires user role: ADMIN

Example:

```
xCommand UserManagement RemoteSupportUser Delete
```

xCommand UserManagement RemoteSupportUser DisablePermanently

Disable the creation of new remote support users. To enable the remote support user again you must factory reset your system.

Requires user role: ADMIN

Parameters:

Confirm(r): <Yes>

Example:

```
xCommand UserManagement RemoteSupportUser DisablePermanently Confirm: Yes
```

xCommand UserManagement RemoteSupportUser GetState

Retrieves the state of the generated remote support user, if one exists.

Requires user role: ADMIN

Example:

```
xCommand UserManagement RemoteSupportUser GetState
```

Video commands

xCommand Video AutoPresentationStart Disable

Not in use in this software version.

Requires user role: ADMIN

Example:

```
xCommand Video AutoPresentationStart Disable
```

xCommand Video AutoPresentationStart Enable

Not in use in this software version.

Requires user role: ADMIN

Example:

```
xCommand Video AutoPresentationStart Enable
```

xCommand Video Input SetMainVideoSource

Set which input source is the main video source. You can identify the source either by the identifier of the connector that it is connected to; or by the identifier of the source itself.

This command requires the use of one, and only one, of these parameters: ConnectorId or SourceId.

ConnectorId: The identifier (ID) of the connector. Connector [n] has ID n. Run the following API command to find the ID: xStatus Video Input Connector. The connector ID is also printed on the codec connector panel.

SourceId: The identifier (ID) of the input source. Input Source [n] has ID n. Run the following API command to find the ID: xStatus Video Input Source.

Requires user role: USER

Parameters:

ConnectorId: <1..5>

SourceId: <1..4>

Example:

```
xCommand Video Input SetMainVideoSource ConnectorId: 1
```

xCommand Video Input Source SetActiveConnector

Input Connector 4 (DVI-I) and Connector 5 (S-Video/Composite) cannot be active simultaneously. Use this command to select which connector is active.

ConnectorId: The identifier (ID) of the connector. Connector [n] has ID n. Run the following API command to find the ID: xStatus Video Input Connector. The connector ID is also printed on the codec connector panel.

Requires user role: USER

Parameters:

ConnectorId: <4..5>

Example:

```
xCommand Video Input Source SetActiveConnector ConnectorId: 4
```

xCommand Video Layout Add

Add a new empty video layout composition. Returns the LayoutId. When the command has been issued you can use the result, the LayoutId of the newly created layout, to add frames with different video sources to the layout.

Requires user role: ADMIN

Parameters:

LayoutId: <1..2147483647>

Example:

```
xCommand Video Layout Add LayoutId: 1
```

xCommand Video Layout Assign

Assign an existing layout to any local or remote output. The defined video composition appears on the specified local output or in the specified remote video stream.

CallId: A unique identifier for the 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.

OutputId: A unique identifier for the local output, which was created when the xCommand Audio LocalOutput Add command was issued.

LayoutId: A unique identifier for the layout, which was created when the xCommand Video Layout Add command was issued.

Requires user role: ADMIN

Parameters:

CallId(r): <0..65534>
OutputId(r): <0..65534>
LayoutId(r): <1..2147483647>

Example:

```
xCommand Video Layout Assign CallId:1 OutputId:1 LayoutId:2
```

xCommand Video Layout AssignCall

Assign the call layout (main stream) to the remote output. The defined video composition appears on the remote main video stream.

LayoutId: A unique identifier for the layout, which was created when the xCommand Video Layout Add command was issued.

CallId: A unique identifier for the 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: ADMIN

Parameters:

LayoutId(r): <1..2147483647>
CallId: <0..65534>

Example:

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

xCommand Video Layout AssignLocalOutput

Assign an existing layout to a local output. The defined video composition appears on the specified local output.

LayoutId: A unique identifier for the layout, which was created when the xCommand Video Layout Add command was issued.

OutputId: A unique identifier for the local output, which was created when the xCommand Audio LocalOutput Add command was issued.

Requires user role: ADMIN

Parameters:

LayoutId(r): <1..2147483647>
OutputId(r): <0..65534>

Example:

```
xCommand Video Layout AssignLocalOutput OutputId:1 LayoutId:2
```

xCommand Video Layout AssignPresentation

Assign the presentation layout (dual stream) to the remote output. The defined video composition appears on the remote dual video stream.

LayoutId: A unique identifier for the layout, which was created when the xCommand Video Layout Add command was issued.

Requires user role: ADMIN

Parameters:

LayoutId(r): <1..2147483647>

Example:

```
xCommand Video Layout AssignPresentation LayoutId:2
```

xCommand Video Layout Frame Add

Add a Video frame to an existing layout. Returns the FrameId. Select size and position of the frame, and the video source to be shown in the frame.

LayoutId: A unique identifier for the layout, which was created when the xCommand Video Layout Add command was issued.

FrameId: A unique identifier of the frame.

PositionX: The top-left X position of the frame.

PositionY: The top-left Y position of the frame.

Width: The width of the frame.

Height: The height of the frame.

Layer: The stacking order of the frames (1-6). Must be unique per frame per layout. Layer 6 is on top.

Border: Select whether or not the frame should have a border.

VideoSourceType: Select the video source to be used as the content of the frame. The parameter arguments are explained below:

Graphic: Not in use.

LocalInput: Select one of the local input sources.

LocalMain: Select what is currently the main source on the local side.

LocalPresentation: Select what is currently the default presentation source on the local side.

MostSpeaking: Select the current loudest speaker in the conference.

OtherMain: Select another site in a call. This involves logic in the codec to ensure that a site does not see itself. You can add several frames with OtherMain. The source and the layout engine automatically populate the frame content with one of the other sites main video, making sure that you do not see the same site more than once, nor yourself.

otherMainRemote:Select another remote site in a call. This involves logic in the codec to ensure that a site does not see itself. You can add several frames with otherMainRemote. The source and the layout engine automatically populate the frame content with one of the other remote sites main video, making sure that you do not see the same site more than once, nor yourself.

OwnMain: Select that each site can see its own self-view.

Presentation: Select what is currently the presentation source (there is always only one active H.239 content in a conference).

RemoteMain: Select the main source of remote site.The CallId must then be specified in the VideoSourceId parameter.

RemotePresentation: Select the presentation source of remote site. The CallId must then be specified in the VideoSourceId parameter.

VideoFile: Not in use.

VideoSourceId: A unique identifier of the video source.

If the VideoSourceType is set to RemoteMain the VideoSourceId is the CallId of the remote site.

If the VideoSourceType is set to LocalInput the VideoSourceId is the video input SourceId.

Otherwise set to 1.

Requires user role: ADMIN

Parameters:

LayoutId(r): <1..2147483647>

FrameId: <1..65535>

PositionX: <0..10000>

PositionY: <0..10000>

Width: <1..10000>

Height: <1..10000>

Layer: <1..6>

Border: <On/Off>

VideoSourceType: <graphic/localInput/localMain/localPresentation/mostSpeaking/mostSpeak-
ing_orderedByPartNumb/otherMain/otherMainRemote/ownMain/participantNumber/presentation/
remoteMain/remotePresentation/videoFile>

VideoSourceId: <0..2147483647>

Example:

```
xCommand Video Layout Frame Add LayoutId:1 PositionX:100 PositionY:100
Width:9800 Height:9800 Layer:1 Border:off VideoSourceType:localInput
VideoSourceId:1
```

xCommand Video Layout Frame List

Lists the data related to all layout frames created by command 'xCommand Video Layout Frame
Add' for a specific layout identified by 'LayoutId'

Requires user role: ADMIN

Parameters:

LayoutId: <1..2147483647>

Example:

```
xCommand Video Layout Frame List LayoutId: 1
```

xCommand Video Layout Frame Remove

Remove a video frame from an existing layout.

LayoutId: A unique identifier for the layout, which was created when the xCommand Video Layout
Add command was issued.

FrameId: A unique identifier of the frame, which was created when the xCommand Video Frame Add
command was issued.

Requires user role: ADMIN

Parameters:

LayoutId(r): <1..2147483647>

FrameId(r): <1..65535>

Example:

```
xCommand Video Layout Frame Remove LayoutId:1 FrameId:1
```

xCommand Video Layout Frame Update

Change the details of an existing frame in a video layout.

LayoutId: A unique identifier for the layout, which was created when the xCommand Video Layout Add command was issued.

FrameId: A unique identifier of the frame, which was created when the xCommand Video Frame Add command was issued.

PositionX: The top-left X position of the frame.

PositionY: The top-left Y position of the frame.

Width: The width of the frame.

Height: The height of the frame.

Layer: The stacking order of the frames (1-6). Must be unique per frame per layout. Layer 6 is on top.

Border: Select whether or not the frame should have a border.

VideoSourceType: Select the video source to be used as the content of the frame. The parameter arguments are explained below:

Graphic: Not in use.

LocalInput: Select one of the local input sources.

LocalMain: Select what is currently the main source on the local side.

LocalPresentation: Select what is currently the default presentation source on the local side.

MostSpeaking: Select the current loudest speaker in the conference.

OtherMain: Select another site in a call. This involves logic in the codec to ensure that a site does not see itself. You can add several frames with OtherMain. The source and the layout engine automatically populate the frame content with one of the other sites main video, making sure that you do not see the same site more than once, nor yourself.

otherMainRemote:Select another remote site in a call. This involves logic in the codec to ensure that a site does not see itself. You can add several frames with otherMainRemote. The source and the layout engine automatically populate the frame content with one of the other remote sites main video, making sure that you do not see the same site more than once, nor yourself.

OwnMain: Select that each site can see its own self-view.

Presentation: Select what is currently the presentation source (there is always only one active H.239 content in a conference).

RemoteMain: Select the main source of remote site.The CallId must then be specified in the VideoSourceId parameter.

RemotePresentation: Select the presentation source of remote site. The CallId must then be specified in the VideoSourceId parameter.

VideoFile: Not in use.

VideoSourceId: A unique identifier of the video source.

If the VideoSourceType is set to RemoteMain the VideoSourceId is the CallId of the remote site.

If the VideoSourceType is set to LocalInput the VideoSourceId is the video input SourceId.

Otherwise set to 1.

Requires user role: ADMIN

Parameters:

LayoutId(r): <1..2147483647>

FrameId(r): <1..65535>

PositionX: <0..10000>

PositionY: <0..10000>

Width: <1..10000>

Height: <1..10000>

Layer: <1..6>

Border: <On/Off>

VideoSourceType: <graphic/localInput/localMain/localPresentation/mostSpeaking/mostSpeaking_orderedByPartNumb/otherMain/otherMainRemote/ownMain/participantNumber/presentation/remoteMain/remotePresentation/videoFile>

VideoSourceId: <0..2147483647>

Example:

```
xCommand Video Layout Frame Update LayoutId:1 FrameId:1
PositionX:200 PositionY:200 Width:9600 Height:9600 Layer:1 Border:off
VideoSourceType:localInput VideoSourceId:1
```

xCommand Video Layout List

Lists all identification numbers for layouts created by command 'xCommand Video Layout Add'.

Requires user role: ADMIN

Example:

```
xCommand Video Layout List
OK
*r VideoLayoutListResult LayoutId: 1
*r VideoLayoutListResult LayoutId: 3
```

xCommand Video Layout LoadDb

Loads and starts using the specified video layout database. The default video layout database is initially provided by the system. The custom database is generated by the Cisco TC Console tool and is made available to the codec from within the TC Console tool. The TC Console software is found at the Developer Zone web page. Go to: http://developer.cisco.com/web/telepresence-developer/tools/integrators/tc-console.

Custom: The system uses the custom video layout database which generated by the Cisco TC Console tool.

CustomAutoModes: The system uses the auto mode part of the custom video layout database.

Default: The system use the default video layout database which is provided by the system.

Requires user role: ADMIN

Parameters:

Type(r): <Custom/CustomAutoModes/Default>

Example:

xCommand Video Layout LoadDb Type: Default

xCommand Video Layout Remove

Remove an existing video layout.

LayoutId: A unique identifier for the layout, which was created when the xCommand Video Layout Add command was issued.

Requires user role: ADMIN

Parameters:

LayoutId(r): <1..2147483647>

Example:

xCommand Video Layout Remove LayoutId: 1

xCommand Video Layout RemoveAll

Remove all existing video layouts.

Requires user role: ADMIN

Example:

xCommand Video Layout RemoveAll

xCommand Video Layout Reset

Reset all layout compositions to factory default.

Requires user role: ADMIN

Example:

xCommand Video Layout Reset

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

xCommand Video Layout UnAssign

Remove the defined video layout, and go back to default.

CallId: A unique identifier for the 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.

OutputId: A unique identifier for the local output, which was created when the xCommand Audio LocalOutput Add command was issued.

Requires user role: ADMIN

Parameters:

CallId(r): <0..65534>
OutputId(r): <0..65534>

Example:

xCommand Video Layout UnAssign CallId: 1 OutputId: 1

xCommand Video Layout UnAssignCall

Remove the defined video layout, and go back to default.

CallId: A unique identifier for the 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: ADMIN

Parameters:

CallId: <0..65534>

Example:

```
xCommand Video Layout UnAssignCall CallId:1
```

xCommand Video Layout UnAssignLocalOutput

Remove the defined video layout, and go back to default.

OutputId: A unique identifier for the local output, which was created when the xCommand Audio LocalOutput Add command was issued.

Requires user role: ADMIN

Parameters:

OutputId(r): <0..65534>

Example:

```
xCommand Video Layout UnAssignLocalOutput OutputId:1
```

xCommand Video Layout UnAssignPresentation

Remove the defined video layout, and go back to default.

Requires user role: ADMIN

Example:

```
xCommand Video Layout UnAssignPresentation
```

xCommand Video Matrix Assign

xCommand Video Matrix commands are a smart overlay to the xCommand Video Layout commands to make it easy to do simple video compositions.

Mode: Choose whether to replace the existing source on that output rendered full screen or to add it. Using Add, the layout engine will recompose the multiple sources automatically. Replace is the default value.

Output: Choose the output you wish to use for this layout.

SourceId: The identifier (ID) of the input source. Input Source [n] has ID n. Run the following API command to find the ID: xStatus Video Input Source.

Requires user role: USER

Parameters:

Mode: <Add / Replace>
Output (r): <1 – 2>
SourceId (r): <1 – 3>

Example:

```
xCommand Video Matrix Assign Mode:Add Output:1 SourceId:1
```

xCommand Video Matrix Reset

Reset the content on the output to the default layout

xCommand Video Matrix commands are a smart overlay to the xCommand Video Layout commands to make it easy to do simple video compositions.

Output: Choose the output you want to reset.

Requires user role: USER

Parameters:

Output: <1 – 2>

Example:

```
xCommand Video Matrix Reset Output:1
```

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

138

Copyright © 2015 Cisco Systems, Inc. All rights reserved.

xCommand Video Matrix Swap

Swap the content defined with xCommand Video Matrix Assign between two outputs.

xCommand Video Matrix commands are a smart overlay to the xCommand Video Layout commands to make it easy to do simple video compositions.

OutputA: The output you are swapping from.

OutputB: The output you are swapping to.

Requires user role: USER

Parameters:

OutputA (r): <1 – 2>

OutputB (r): <1 – 2>

Example:

```
xCommand Video Matrix Swap OutputA:1 OutputB:2
```

xCommand Video Matrix Unassign

Remove a source from an output. Just as with xCommand Video Matrix Assign the layout engine will recompose the remaining sources automatically.

xCommand Video Matrix commands are a smart overlay to the xCommand Video Layout commands to make it easy to do simple video compositions.

Output: Choose the output you wish to remove the source from.

SourceId: The identifier (ID) of the input source. Input Source [n] has ID n. Run the following API command to find the ID: xStatus Video Input Source.

Requires user role: USER

Parameters:

Output (r): <1 – 2>

SourceId (r): <1 – 3>

Example:

```
xCommand Video Matrix Unassign Output:1 SourceId:1
```

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
```

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
```

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
```

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
```

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
```

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: <On/Off>

FullscreenMode: <On/Off>

PIPPosition: <CenterLeft/CenterRight/LowerLeft/LowerRight/UpperCenter/UpperLeft/ UpperRight>

OnMonitorRole: <First/Fourth/Second/Third>

Example:

```
Example 1
xCommand Video Selfview Set Mode: On FullscreenMode: Off PIPPosition:
CenterLeft OnMonitorRole: First


Example 2
xCommand Video Selfview Set FullscreenMode: On OnMonitorRole: Second
```

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

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

142

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/sx-docs

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 VolumeMute

Shows whether the endpoint volume is set to mute.

Value space of the result returned:

<Off/On>

Example:

```
xStatus Audio VolumeMute
*s Audio VolumeMute: Off
** end
```

xStatus Audio Input LocalInput [n] Name

Shows the name of the local input. A local input is a mix of input connectors. You can run the command xStatus Audio Input LocalInput to find the identity [n] of the input.

Value space of the result returned:

<String>

Example:

```
xStatus Audio Input LocalInput 1 Name
*s Audio Input LocalInput 1 Name: "Microphone"
** end
```

xStatus Audio Input LocalInput [n] MixerMode

Shows how the local input connectors have been mixed together. You can run the command xStatus Audio Input LocalInput to find the identity [n] of the input.

Auto: The microphone with the strongest speaker is active and the others are strongly attenuated.

Fixed: The input connector signals are mixed together with equal gains.

GainShared: The microphones are given a normalized gain factor relative to the strongest speaker before being mixed together.

Value space of the result returned:

<"Auto"/"Fixed"/"GainShared">

Example:

```
xStatus Audio Input LocalInput 1 MixerMode
*s Audio Input LocalInput 1 MixerMode: "Auto"
** end
```

xStatus Audio Input LocalInput [n] Mute

Shows the mute mode for the local inputs. A local input is a mix of input connectors.

You can run the command xStatus Audio Input LocalInput to find the identity [n] of the input.

Value space of the result returned:

<"On"/"Off">

Example:

```
xStatus Audio Input LocalInput 1 Mute
*s Audio Input LocalInput 1 Mute: "Off"
** end
```

xStatus Audio Input LocalInput [n] Channels

States if the local input channels are mixed into into a mono signal (1) or stereo signal (2).

You can run the command xStatus Audio Input LocalInput to find the identity [n] of the input.

Value space of the result returned:

<1..2>

Example:

```
xStatus Audio Input LocalInput 1 Channels
*s Audio Input LocalInput 1 Channels: 1
** end
```

xStatus Audio Input LocalInput [n] AGC

Shows the AGC (Automatic Gain Control) mode on the local input.

You can run the command xStatus Audio Input LocalInput to find the identity [n] of the input.

Value space of the result returned:

<"On"/"Off">

Example:

```
xStatus Audio Input LocalInput 1 AGC
*s Audio Input LocalInput 1 AGC: "On"
** end
```

xStatus Audio Input LocalInput [n] Connector

Lists the connectors that are attached to the local input.

You can run the command xStatus Audio Input LocalInput to find the identity [n] of the input.

Value space of the result returned:

<"Microphone.1"/../"Microphone.4"/"Line.1"/"Line.2"/"HDMI.2">

Example:

```
xStatus Audio Input LocalInput 1 Connector
*s Audio Input LocalInput 1 Connector: "Microphone.1"
** end
```

xStatus Audio Input RemoteInput [n] CallId

Shows the CallId for the remote audio input.

You can run the command xStatus Audio Input RemoteInput to find the identity [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 [n] EcReferenceDelay

Returns the detected latency for each loudspeaker to microphone path. The result is in milliseconds.

Value space of the result returned:

<Integer>

Example:

```
xStatus Audio Input Connectors Microphone 1 EcReferenceDelay
*s Audio Connectors Microphone 1 EcReferenceDelay: 120
** end
```

xStatus Audio Output LocalOutput [n] Name

Shows the name of the local output.

You can run the command xStatus Audio Output LocalOutput to find the identity [n] of the output.

Value space of the result returned:

<String>

Example:

```
xStatus Audio Output LocalOutput 4 Name
*s Audio Output LocalOutput 4 Name: "MyLocalOutput1"
** end
```

xStatus Audio Output LocalOutput [n] Loudspeaker

Shows the Loudspeaker mode on the local output. If one or more of the output connectors that are attached to the local output are connected to a loudspeaker, then this signal should be a reference signal to the echo canceler and Loudspeaker should be set to On.

You can run the command xStatus Audio Output LocalOutput to find the identity [n] of the output.

Value space of the result returned:

<"On"/"Off">

Example:

xStatus Audio Output LocalOutput 4 Loudspeaker

*s Audio Output LocalOutput 4 Loudspeaker: "Off"

** end

xStatus Audio Output LocalOutput [n] Channels

States if the local output channels are mixed into into a mono signal (1) or stereo signal (2).

You can run the command xStatus Audio Output LocalOutput to find the identity [n] of the output.

Value space of the result returned:

<1..2>

Example:

xStatus Audio Output LocalOutput 4 Channels

*s Audio Output LocalOutput 4 Channels: "1"

** end

xStatus Audio Output LocalOutput [n] Connector

Lists the connectors that are attached to the local output.

You can run the command xStatus Audio Output LocalOutput to find the identity [n] of the output.

Value space of the result returned:

<"Line.1"/"Line.2"/"HDMI.1"/"HDMI.2">

Example:

xStatus Audio Output LocalOutput 4 Connector

*s Audio Output LocalOutput 4 Connector: "Line.1"

** end

xStatus Audio Output LocalOutput [n] Input [n] Gain

Shows the gain (dB) on the input, when it is connected to the local output. Range from -54 dB to 15 dB, where -54 dB equals Off.

You can run the command xStatus Audio Output LocalOutput to find the identity [n] of the output and input.

Value space of the result returned:

<-54..15>

Example:

xStatus Audio Output LocalOutput 4 Input 2 Gain

*s Audio Output LocalOutput 4 Input 2 Gain: 0

** end

xStatus Audio Output LocalOutput [n] VolumeControlled

Shows whether the volume of this output is controlled by global audio control (Off), or if it is at the level defined with xCommand Audio Volume Set (On).

Value space of the result returned:

<String>

Example:

xStatus Audio Output LocalOutput 5 VolumeControlled

*s Audio Output LocalOutput 5 VolumeControlled: "On"

** end

xStatus Audio Output RemoteOutput [n] CallId

Shows the CallId for the remote audio output.

You can run the command xStatus Audio Output RemoteOutput to find the identity [n] of the output.

Value space of the result returned:

<1..65534>

Example:

xStatus Audio Output RemoteOutput 9 CallId

*s Audio Output RemoteOutput 9 CallId: 28

** end

xStatus Audio Output RemoteOutput [n] Input [n] Gain

Shows the gain (dB) on the input, when input is connected to the remote output. Range from -54 dB to 15 dB, where -54 dB equals Off.

You can run the command xStatus Audio Output RemoteOutput to find the identity [n] of the output and input.

Value space of the result returned:

<-54..15>

Example:

```
xStatus Audio Output RemoteOutput 9 Input 1 Gain
*s Audio Output RemoteOutput 9 Input 1 Gain: 0
** 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 [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 [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 [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:

<String>

Example:

```
xStatus Call 27 Protocol
*s Call 27 Protocol: "h323"
** end
```

xStatus Call [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 [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 [n] CallbackNumber

Shows the remote (far end) number or URI of an incoming or outgoing call, including the call proto-col, 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 [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 [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 [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 [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 [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 [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 [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 [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 [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 [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 [n] DeviceType

Indicates whether the remote system is a single endpoint or an MCU. Some Cisco endpoints (EX90, C40, C60, C90, SX20, SX80, MX200 G2, MX300 G2) have built-in MCU capabilities.

Value space of the result returned:

<Endpoint/MCU>

Example:

```
xStatus Call DeviceType
*s Call 4 DeviceType: Endpoint
** end
```

xStatus Call [n] AttendedTransferFrom

Shows the CallId for the call the current call was transferred from.

Value space of the result returned:

<Integer>

Example:

```
xStatus Call 1 AttendedTransferFrom
*s Call 1 AttendedTransferFrom: 1234
** end
```

Camera status

xStatus Camera

Shows the top level overview of the camera status.

xStatus Camera [n] 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 [n] 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 [n] 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
```

157

xStatus Camera [n] 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 [n] 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 [n] 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 [n] 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 [n] 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 [n] 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 [n] 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 [n] 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 [n] 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 [n] 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 [n] 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
```

xStatus Camera [n] UpgradeStatus

Returns the status of the camera software upgrade.

None: No upgrade going on at the moment.

Downloading: Downloading the upgrade.

Installing: Installing the upgrade.

Rebooting: Rebooting the system to take the upgrade in use.

Value space of the result returned:

<None/Downloading/Installing/Rebooting>

Example:

```
xStatus Camera 1 UpgradeStatus
*s Camera 1 UpgradeStatus: None
** end
```

xStatus Camera [n] DownloadProgress

Indicates how many percent of the new camera software upgrade has been downloaded. Result –1 indicates that the information is not available for that camera type.

Value space of the result returned:

<Integer>

Example:

```
xStatus Camera 1 DownloadProgress
*s Camera 1 DownloadProgress: 0
** end
```

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

159

Copyright © 2015 Cisco Systems, Inc. All rights reserved.

Cameras status

xStatus Cameras SpeakerTrack Availability

Shows whether speaker tracking is available.

Off: Speaker tracking is turned off with the command xConfiguration Cameras SpeakerTrack Mode: Off

Unavailable: Speaker tracking hardware is not found.

Available: Speaker tracking hardware and the two cameras are found, and it is possible to turn speaker tracking on or off.

Value space of the result returned:

<Off/Unavailable/Available>

Example:

```
xStatus Cameras SpeakerTrack Availability
*s Cameras SpeakerTrack Availability: Unavailable
** end
```

xStatus Cameras SpeakerTrack LeftCamera VideoInputConnector

Returns the number of the HDMI connector used for the left camera on a SpeakerTrack camera.

Value space of the result returned:

<0/1/2/3>

Example:

```
xStatus Cameras SpeakerTrack LeftCamera VideoInputConnector
*s Cameras SpeakerTrack LeftCamera VideoInputConnector: 0
** end
```

xStatus Cameras SpeakerTrack RightCamera VideoInputConnector

Returns the number of the HDMI connector used for the right camera on a SpeakerTrack camera.

Value space of the result returned:

<0/1/2/3>

Example:

```
xStatus Cameras SpeakerTrack RightCamera VideoInputConnector
*s Cameras SpeakerTrack RightCamera VideoInputConnector: 0
** end
```

xStatus Cameras SpeakerTrack Status

Shows whether speaker tracking is set to active or inactive.

Value space of the result returned:

<Active/Inactive>

Example:

```
xStatus Cameras SpeakerTrack Status
*s Cameras SpeakerTrack Status: Active
** end
```

xStatus Cameras SpeakerTrack Whiteboard [n] Camera [n] Defined

Shows whether a snap to whiteboard position has been stored with xCommand Cameras SpeakerTrack Whiteboard StorePosition for the specified whiteboard and camera.

Value space of the result returned:

<False/True>

Example:

```
xStatus Cameras SpeakerTrack Whiteboard 1 Camera 1  Defined
*s Cameras SpeakerTrack Whiteboard 1 Camera 1 Defined: False
** end
```

xStatus Cameras SpeakerTrack Whiteboard [n] Camera [n] Pan

Shows the pan position stored with xCommand Cameras SpeakerTrack Whiteboard StorePosition for the specified whiteboard and camera.

Value space of the result returned:

<Integer>

Example:

```
xStatus Cameras SpeakerTrack Whiteboard 1 Camera 1 Pan
*s Cameras SpeakerTrack Whiteboard 1 Camera 1 Pan: 0
** end
```

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

160

Copyright © 2015 Cisco Systems, Inc. All rights reserved.

xStatus Cameras SpeakerTrack Whiteboard [n] Camera [n] Tilt

Shows the tilt position stored with xCommand Cameras SpeakerTrack Whiteboard StorePosition for the specified whiteboard and camera.

Value space of the result returned:

<Integer>

Example:

```
xStatus Cameras SpeakerTrack Whiteboard 1 Camera 1 Tilt
*s Cameras SpeakerTrack Whiteboard 1 Camera 1 Tilt: 0
** end
```

xStatus Cameras SpeakerTrack Whiteboard [n] Camera [n] Zoom

Shows the zoom position stored with xCommand Cameras SpeakerTrack Whiteboard StorePosition for the specified whiteboard and camera.

Value space of the result returned:

<Integer>

Example:

```
xStatus Cameras SpeakerTrack Whiteboard 1 Camera 1 Zoom
*s Cameras SpeakerTrack Whiteboard 1 Camera 1 Zoom: 4188
** end
```

xStatus Cameras SpeakerTrack Whiteboard [n] Distance

Shows the distance set with xCommand Cameras SpeakerTrack Whiteboard SetDistance for the specified whiteboard.

Value space of the result returned:

<Integer>

Example:

```
xStatus Cameras SpeakerTrack Whiteboard 1 Distance
*s Cameras SpeakerTrack Whiteboard 1 Distance: -1
** 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:

<Off/Sending/Receiving>

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 Presentation Instance [n] LocalSendingMode

Shows whether a presentation source is shared locally or with a remote participant. There can be multiple local presentations which all have their own instance.

Value space of the result returned:

<LocalOnly/LocalRemote/Off>

Example:

xStatus Conference Presentation Instance 1 LocalSendingMode

*s Conference Presentation Instance 1 LocalSendingMode: LocalOnly

** end

xStatus Conference Presentation Instance [n] LocalSource

Shows the SourceId for a current presentation. There can be multiple local presentations which all have their own instance.

Value space of the result returned:

<Integer>

Example:

xStatus Conference Presentation Instance 1 LocalSource

*s Conference Presentation Instance 1 LocalSource: 1

** end

xStatus Conference Site [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 [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 [n] Capabilities FECC Source [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 [n] Capabilities FECC Source [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 [n] Capabilities FECC Source [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 [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 [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 [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 [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 [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 [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 [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 [n] BookingId

Shows the booking ID of a conference (if assigned). The booking ID can be used for easy identifica-
tion 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 [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/Encrypted>

Example:

```
xStatus Conference Site SecurityStatus
*s Conference Site 21 SecurityStatus: Unknown
** end
```

xStatus Conference Site [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 [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 [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 [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 [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 [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 [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 [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 [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 [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 [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/MultiSite/Auto/CUCMMediaResourceGroupList>

Example:

xStatus Conference Multipoint Mode

*s Conference Multipoint Mode: "MultiSite"

** 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

Shows 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
```

xStatus Conference ActiveSpeaker Manual SiteId

Shows the SiteId (CallId) for the participant locked as the active speaker in the conference.

Value space of the result returned:

<Integer>

Example:

```
xStatus Conference ActiveSpeaker Manual SiteId
*s Conference ActiveSpeaker Manual SiteId: 0
** end
```

xStatus Conference SelectedCallProtocol

Indicates which call protocol is used as default.

Value space of the result returned:

<H323/SIP/H320>

Example:

```
xStatus Conference SelectedCallProtocol
*s Conference SelectedCallProtocol: SIP
** 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 [n] Channels IncomingAudioChannel [n] Netstat 1 Jitter

xStatus Diagnostics Call [n] Channels IncomingVideoChannel [n] Netstat 1 Jitter

xStatus Diagnostics Call [n] Channels IncomingDataChannel [n] Netstat 1 Jitter

xStatus Diagnostics Call [n] Channels OutgoingAudioChannel [n] Netstat 1 Jitter

xStatus Diagnostics Call [n] Channels OutgoingVideoChannel [n] Netstat 1 Jitter

xStatus Diagnostics Call [n] Channels OutgoingDataChannel [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 [n] Channels IncomingAudioChannel [n] Netstat 1 Packets

xStatus Diagnostics Call [n] Channels IncomingVideoChannel [n] Netstat 1 Packets

xStatus Diagnostics Call [n] Channels IncomingDataChannel [n] Netstat 1 Packets

xStatus Diagnostics Call [n] Channels OutgoingAudioChannel [n] Netstat 1 Packets

xStatus Diagnostics Call [n] Channels OutgoingVideoChannel [n] Netstat 1 Packets

xStatus Diagnostics Call [n] Channels OutgoingDataChannel [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 [n] Channels IncomingAudioChannel [n] Netstat 1 Loss

xStatus Diagnostics Call [n] Channels IncomingVideoChannel [n] Netstat 1 Loss

xStatus Diagnostics Call [n] Channels IncomingDataChannel [n] Netstat 1 Loss

xStatus Diagnostics Call [n] Channels OutgoingAudioChannel [n] Netstat 1 Loss

xStatus Diagnostics Call [n] Channels OutgoingVideoChannel [n] Netstat 1 Loss

xStatus Diagnostics Call [n] Channels OutgoingDataChannel [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 [n] Channels IncomingAudioChannel [n] Netstat 1 LastIntervalLost

xStatus Diagnostics Call [n] Channels IncomingVideoChannel [n] Netstat 1 LastIntervalLost

xStatus Diagnostics Call [n] Channels IncomingDataChannel [n] Netstat 1 LastIntervalLost

xStatus Diagnostics Call [n] Channels OutgoingAudioChannel [n] Netstat 1 LastIntervalLost

xStatus Diagnostics Call [n] Channels OutgoingVideoChannel [n] Netstat 1 LastIntervalLost

xStatus Diagnostics Call [n] Channels OutgoingDataChannel [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 [n] Channels IncomingAudioChannel [n] Netstat 1 LastIntervalReceived

xStatus Diagnostics Call [n] Channels IncomingVideoChannel [n] Netstat 1 LastIntervalReceived

xStatus Diagnostics Call [n] Channels IncomingDataChannel [n] Netstat 1 LastIntervalReceived

xStatus Diagnostics Call [n] Channels OutgoingAudioChannel [n] Netstat 1 LastIntervalReceived

xStatus Diagnostics Call [n] Channels OutgoingVideoChannel [n] Netstat 1 LastIntervalReceived

xStatus Diagnostics Call [n] Channels OutgoingDataChannel [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 [n] Channels IncomingAudioChannel [n] Netstat 1 Drop

xStatus Diagnostics Call [n] Channels IncomingVideoChannel [n] Netstat 1 Drop

xStatus Diagnostics Call [n] Channels IncomingDataChannel [n] Netstat 1 Drop

xStatus Diagnostics Call [n] Channels OutgoingAudioChannel [n] Netstat 1 Drop

xStatus Diagnostics Call [n] Channels OutgoingVideoChannel [n] Netstat 1 Drop

xStatus Diagnostics Call [n] Channels OutgoingDataChannel [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 [n] Channels IncomingAudioChannel [n] Netstat 1 Bytes

xStatus Diagnostics Call [n] Channels IncomingVideoChannel [n] Netstat 1 Bytes

xStatus Diagnostics Call [n] Channels IncomingDataChannel [n] Netstat 1 Bytes

xStatus Diagnostics Call [n] Channels OutgoingAudioChannel [n] Netstat 1 Bytes

xStatus Diagnostics Call [n] Channels OutgoingVideoChannel [n] Netstat 1 Bytes

xStatus Diagnostics Call [n] Channels OutgoingDataChannel [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 [n] Channels IncomingAudioChannel [n] Netstat 1 ChannelRate

xStatus Diagnostics Call [n] Channels IncomingVideoChannel [n] Netstat 1 ChannelRate

xStatus Diagnostics Call [n] Channels IncomingDataChannel [n] Netstat 1 ChannelRate

xStatus Diagnostics Call [n] Channels OutgoingAudioChannel [n] Netstat 1 ChannelRate

xStatus Diagnostics Call [n] Channels OutgoingVideoChannel [n] Netstat 1 ChannelRate

xStatus Diagnostics Call [n] Channels OutgoingDataChannel [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 [n] Channels IncomingAudioChannel [n] Netstat 1 MaxJitter

xStatus Diagnostics Call [n] Channels IncomingVideoChannel [n] Netstat 1 MaxJitter

xStatus Diagnostics Call [n] Channels IncomingDataChannel [n] Netstat 1 MaxJitter

xStatus Diagnostics Call [n] Channels OutgoingAudioChannel [n] Netstat 1 MaxJitter

xStatus Diagnostics Call [n] Channels OutgoingVideoChannel [n] Netstat 1 MaxJitter

xStatus Diagnostics Call [n] Channels OutgoingDataChannel [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
```

GPIO status

xStatus GPIO Pin [1..4] State
NOTE: This command is not supported on Codec C40.

Shows the current state of each GPIO pin. The default state is High (+12V). When activated the state is Low (0V).

Value space of the result returned:

<High/Low>

Example:

```
xStatus GPIO Pin 1 State
*s GPIO Pin 1 State: High
** end
```

H320 status

xStatus H320
Shows the top level overview of the 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
```

xStatus H323 Mode Status
Shows the status for H.323 registration.
Enabled: Registration is enabled.
Disabled: Registration ifs disable, because SIP is enabled.
NotAvailable: Status is not available.

Value space of the result returned:
<String>

Example:
```
xStatus H323 Mode Status
*s H323 Mode Status: "Disabled"
** end
```

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

174

HttpFeedback status

ICE 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>

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
```

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

175

Copyright © 2015 Cisco Systems, Inc. All rights reserved.

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 Turn Transport

Shows which transport protocol is used.

Value space of the result returned:

<UDP/TCP/Unknown>

Example:

```
xStatus ICE Turn Transport
*s ICE Turn Transport: Unknown
** 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
```

xStatus ICE Turn Discovermode

Returns the result of the turn server test, if xConfiguration SIP Profile[1] Turn DiscoverMode is set to On.

Verified: The server was reached, and the user name and password were received.

NotVerified: Configuration has been received, but neither server or credentials has been verified.

Failed: The server was not reached.

Off: Turn discovery mode is set to Off.

Value space of the result returned:

<Verified/NotVerified/Failed/Off>

Example:

```
xStatus ICE Turn Discovermode
*s ICE Turn Discovermode: "Verified"
** end
```

xStatus ICE Turn Bandwidth

Returns the bandwidth available for the turn server right after xConfiguration SIP Profile 1 Turn BandwidthProbe is set to On.

Probing failed: Could not obtain information on bandwidth.

Probe Aborted: The probing was canceled because a call started.

Probe Connected: The probing has stared.

Probe progress: Shows the bandwidth at this stage of the test.

Probe stopped with loss: The probing finished, because packet loss was detected. The loss is shown in kbps.

Probe Completed: The probing completed. The bandwidth result shown in kbps.

Value space of the result returned:

<"Probing failed"/"Probe Aborted"/"Probe Connected"/"Probe progress"/"Probe stopped with loss: kbps"/"Probe Completed: kbps">

Example:

```
xStatus ICE Turn Bandwidth
*s ICE Turn Bandwidth: "Probe Completed: 6000kbps"
** end
```

Logging status

xStatus Logging ExtendedLogging Mode

Indicates whether extended logging of diagnostics is turned on or off.

Value space of the result returned:

<On/Off>

Example:

```
xStatus Logging ExtendedLogging Mode
*s Logging ExtendedLogging Mode: Off
** 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 [n] IncomingAudioChannel [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 [n] IncomingAudioChannel [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.

Opus: Opus is a lossy audio compression format.

Value space of the result returned:

<AACLD/G722/G7221/G711Mu/G711A/G729/G729AB/Opus>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Audio Protocol
*s MediaChannels Call 27 IncomingAudioChannel 327 Audio Protocol: AACLD
** end
```

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

178

xStatus MediaChannels Call [n] IncomingAudioChannel [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 [n] IncomingAudioChannel [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 [n] IncomingAudioChannel [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 [n] IncomingAudioChannel [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 [n] IncomingAudioChannel [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 [n] IncomingAudioChannel [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 [n] IncomingAudioChannel [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 [n] IncomingAudioChannel [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 [n] IncomingAudioChannel [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 [n] IncomingAudioChannel [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 [n] IncomingVideoChannel [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 [n] IncomingVideoChannel [n] ChannelRole

Shows if the incoming channel is the main video channel or presentation channel.

Value space of the result returned:

<Main/Presentation>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 ChannelRole

*s MediaChannels Call 27 IncomingVideoChannel 330 ChannelRole: Main

** end
```

180

xStatus MediaChannels Call [n] IncomingVideoChannel [n] Video Protocol

Shows the video algorithm for the incoming video channel.

HEVC: The H.265 algorithm is an ITU-T standard for video compression.

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:

<H264/H263pp/H263/H261/HEVC>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Video Protocol
*s MediaChannels Call 27 IncomingVideoChannel 330 Video Protocol: H264
** end
```

xStatus MediaChannels Call [n] IncomingVideoChannel [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 [n] IncomingVideoChannel [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 [n] IncomingVideoChannel [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 [n] IncomingVideoChannel [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 [n] IncomingVideoChannel [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
```

xStatus MediaChannels Call [n] IncomingVideoChannel [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 [n] IncomingVideoChannel [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 [n] IncomingVideoChannel [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 [n] IncomingVideoChannel [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 [n] IncomingVideoChannel [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 [n] IncomingVideoChannel [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 [n] OutgoingAudioChannel [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 [n] OutgoingAudioChannel [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.

Opus: Opus is a lossy audio compression format.

Value space of the result returned:

<AACLD/G722/G7221/G711Mu/G711A/G729/G729AB/Opus>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Audio Protocol
*s MediaChannels Call 27 OutgoingAudioChannel 328 Audio Protocol: AACLD
** end
```

xStatus MediaChannels Call [n] OutgoingAudioChannel [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 [n] OutgoingAudioChannel [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 [n] OutgoingAudioChannel [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 [n] OutgoingAudioChannel [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
```

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

183

Copyright © 2015 Cisco Systems, Inc. All rights reserved.

xStatus MediaChannels Call [n] OutgoingAudioChannel [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 [n] OutgoingAudioChannel [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 [n] OutgoingAudioChannel [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 [n] OutgoingAudioChannel [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 [n] OutgoingAudioChannel [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 [n] OutgoingVideoChannel [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 [n] OutgoingVideoChannel [n] ChannelRole

Shows if the outgoing channel is the main video channel or presentation channel.

Value space of the result returned:

<Main/Presentation>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 ChannelRole
*s MediaChannels Call 27 OutgoingVideoChannel 331 ChannelRole: Main
** end
```

xStatus MediaChannels Call [n] OutgoingVideoChannel [n] Video Protocol

Shows the video algorithm for the outgoing video channel.

HEVC: The H.265 algorithm is an ITU–T standard for video compression.

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:

<H264/H263pp/H263/H261/HEVC>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video Protocol
*s MediaChannels Call 27 OutgoingVideoChannel 331 Video Protocol: "H264"
** end
```

xStatus MediaChannels Call [n] OutgoingVideoChannel [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 [n] OutgoingVideoChannel [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 [n] OutgoingVideoChannel [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 [n] OutgoingVideoChannel [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 [n] OutgoingVideoChannel [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 [n] OutgoingVideoChannel [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 [n] OutgoingVideoChannel [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 [n] OutgoingVideoChannel [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 [n] OutgoingVideoChannel [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 [n] OutgoingVideoChannel [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 [n] OutgoingVideoChannel [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
```

xStatus MediaChannels Call [n] IncomingAudioChannel [n] Transport RTCP Local Protocol

Shows the local transport protocol used for RTCP for incoming audio.

The result Unknown is shown when an address is not available, for example during call setup, or for channels that are not supported by the far-end.

Value space of the result returned:

<UDP/TCP/Unknown>

Example:

```
xStatus MediaChannels Call 2  IncomingAudioChannel 127  Transport RTCP
Local Protocol
*s MediaChannels Call 2 IncomingAudioChannel 127 Transport RTCP Local
Protocol: UDP
** end
```

xStatus MediaChannels Call [n] IncomingAudioChannel [n] Transport RTCP Remote Protocol

Shows the remote transport protocol used for RTCP for incoming audio.

The result Unknown is shown when an address is not available, for example during call setup, or for channels that are not supported by the far-end.

Value space of the result returned:

<UDP/TCP/Unknown>

Example:

```
xStatus MediaChannels Call 2 IncomingAudioChannel 127 Transport RTCP
Remote Protocol
*s MediaChannels Call 2 IncomingAudioChannel 127 Transport RTCP Remote
Protocol: UDP
** end
```

xStatus MediaChannels Call [n] IncomingAudioChannel [n] Transport RTP Local Protocol

Shows the local transport protocol used for RTP for incoming audio.

The result Unknown is shown when an address is not available, for example during call setup, or for channels that are not supported by the far-end.

Value space of the result returned:

<UDP/TCP/Unknown>

Example:

```
xStatus MediaChannels Call 2 IncomingAudioChannel 127 Transport RTP Local
Protocol
*s MediaChannels Call 2 IncomingAudioChannel 127 Transport RTP Local
Protocol: UDP
** end
```

xStatus MediaChannels Call [n] IncomingAudioChannel [n] Transport RTP Remote Protocol

Shows the remote transport protocol used for RTP for incoming audio.

The result Unknown is shown when an address is not available, for example during call setup, or for channels that are not supported by the far-end.

Value space of the result returned:

<UDP/TCP/Unknown>

Example:

```
xStatus MediaChannels Call 2 IncomingAudioChannel 127 Transport RTP
Remote Protocol
*s MediaChannels Call 2 IncomingAudioChannel 127 Transport RTP Remote
Protocol: UDP
** end
```

xStatus MediaChannels Call [n] IncomingVideoChannel [n] Transport RTCP Local Protocol

Shows the local transport protocol used for RTCP for incoming video.

The result Unknown is shown when an address is not available, for example during call setup, or for channels that are not supported by the far-end.

Value space of the result returned:

<UDP/TCP/Unknown>

Example:

```
xStatus MediaChannels Call 2 IncomingVideoChannel 127 Transport RTCP
Local Protocol
*s MediaChannels Call 2 IncomingVideoChannel 127 Transport RTCP Local
Protocol: UDP
** end
```

xStatus MediaChannels Call [n] IncomingVideoChannel [n] Transport RTCP Remote Protocol

Shows the remote transport protocol used for RTCP for incoming video.

The result Unknown is shown when an address is not available, for example during call setup, or for channels that are not supported by the far-end.

Value space of the result returned:

<UDP/TCP/Unknown>

Example:

```
xStatus MediaChannels Call 2 IncomingVideoChannel 127 Transport RTCP
Remote Protocol
*s MediaChannels Call 2 IncomingVideoChannel 127 Transport RTCP Remote
Protocol: UDP
** end
```

xStatus MediaChannels Call [n] IncomingVideoChannel [n] Transport RTP Local Protocol

Shows the local transport protocol used for RTP for incoming video.

The result Unknown is shown when an address is not available, for example during call setup, or for channels that are not supported by the far-end.

Value space of the result returned:

<UDP/TCP/Unknown>

Example:

```
xStatus MediaChannels Call 2 IncomingVideoChannel 127 Transport RTP Local
Protocol
*s MediaChannels Call 2 IncomingVideoChannel 127 Transport RTP Local
Protocol: UDP
** end
```

xStatus MediaChannels Call [n] IncomingVideoChannel [n] Transport RTP Remote Protocol

Shows the remote transport protocol used for RTP for incoming video.

The result Unknown is shown when an address is not available, for example during call setup, or for channels that are not supported by the far-end.

Value space of the result returned:

<UDP/TCP/Unknown>

Example:

```
xStatus MediaChannels Call 2 IncomingVideoChannel 127 Transport RTP
Remote Protocol
*s MediaChannels Call 2 IncomingVideoChannel 127 Transport RTP Remote
Protocol: UDP
** end
```

xStatus MediaChannels Call [n] OutgoingAudioChannel [n] Transport RTCP Local Protocol

Shows the local transport protocol used for RTCP for outgoing audio.

The result Unknown is shown when an address is not available, for example during call setup, or for channels that are not supported by the far-end.

Value space of the result returned:

<UDP/TCP/Unknown>

Example:

```
xStatus MediaChannels Call 2 OutgoingAudioChannel 127 Transport RTCP
Local Protocol
*s MediaChannels Call 2 OutgoingAudioChannel 127 Transport RTCP Local
Protocol: UDP
** end
```

188

xStatus MediaChannels Call [n] OutgoingAudioChannel [n] Transport RTCP Remote Protocol

Shows the remote transport protocol used for RTCP for outgoing audio.

The result Unknown is shown when an address is not available, for example during call setup, or for channels that are not supported by the far-end.

Value space of the result returned:

<UDP/TCP/Unknown>

Example:

```
xStatus MediaChannels Call 2 OutgoingAudioChannel 127 Transport RTCP
Remote Protocol
*s MediaChannels Call 2 OutgoingAudioChannel 127 Transport RTCP Remote
Protocol: UDP
** end
```

xStatus MediaChannels Call [n] OutgoingAudioChannel [n] Transport RTP Local Protocol

Shows the local transport protocol used for RTP for outgoing audio.

The result Unknown is shown when an address is not available, for example during call setup, or for channels that are not supported by the far-end.

Value space of the result returned:

<UDP/TCP/Unknown>

Example:

```
xStatus MediaChannels Call 2 OutgoingAudioChannel 127 Transport RTP Local
Protocol
*s MediaChannels Call 2 OutgoingAudioChannel 127 Transport RTP Local
Protocol: UDP
** end
```

xStatus MediaChannels Call [n] OutgoingAudioChannel [n] Transport RTP Remote Protocol

Shows the remote transport protocol used for RTP for outgoing audio.

The result Unknown is shown when an address is not available, for example during call setup, or for channels that are not supported by the far-end.

Value space of the result returned:

<UDP/TCP/Unknown>

Example:

```
xStatus MediaChannels Call 2 OutgoingAudioChannel 127 Transport RTP
Remote Protocol
*s MediaChannels Call 2 OutgoingAudioChannel 127 Transport RTP Remote
Protocol: UDP
** end
```

xStatus MediaChannels Call [n] OutgoingVideoChannel [n] Transport RTCP Local Protocol

Shows the local transport protocol used for RTCP for outgoing video.

The result Unknown is shown when an address is not available, for example during call setup, or for channels that are not supported by the far-end.

Value space of the result returned:

<UDP/TCP/Unknown>

Example:

```
xStatus MediaChannels Call 2 OutgoingVideoChannel 127 Transport RTCP
Local Protocol
*s MediaChannels Call 2 OutgoingVideoChannel 127 Transport RTCP Local
Protocol: UDP
** end
```

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

189

Copyright © 2015 Cisco Systems, Inc. All rights reserved.

xStatus MediaChannels Call [n] OutgoingVideoChannel [n] Transport RTCP Remote Protocol

Shows the remote transport protocol used for RTCP for outgoing video.

The result Unknown is shown when an address is not available, for example during call setup, or for channels that are not supported by the far-end.

Value space of the result returned:

<UDP/TCP/Unknown>

Example:

```
xStatus MediaChannels Call 2 OutgoingVideoChannel 127 Transport RTCP
Remote Protocol
*s MediaChannels Call 2 OutgoingVideoChannel 127 Transport RTCP Remote
Protocol: UDP
** end
```

xStatus MediaChannels Call [n] OutgoingVideoChannel [n] Transport RTP Local Protocol

Shows the local transport protocol used for RTP for outgoing video.

The result Unknown is shown when an address is not available, for example during call setup, or for channels that are not supported by the far-end.

Value space of the result returned:

<UDP/TCP/Unknown>

Example:

```
xStatus MediaChannels Call 2 OutgoingVideoChannel 127 Transport RTP Local
Protocol
*s MediaChannels Call 2 OutgoingVideoChannel 127 Transport RTP Local
Protocol: UDP
** end
```

xStatus MediaChannels Call [n] OutgoingVideoChannel [n] Transport RTP Remote Protocol

Shows the remote transport protocol used for RTP for outgoing video.

The result Unknown is shown when an address is not available, for example during call setup, or for channels that are not supported by the far-end.

Value space of the result returned:

<UDP/TCP/Unknown>

Example:

```
xStatus MediaChannels Call 2 OutgoingVideoChannel 127 Transport RTP
Remote Protocol
*s MediaChannels Call 2 OutgoingVideoChannel 127 Transport RTP Remote
Protocol: UDP
** end
```

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

190

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 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 provi-sioning 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 provi-sioning 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 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 1 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 1 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 1 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 1 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 1 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 1 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 1 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 1 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 1 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 1 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 1 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 1 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
```

xStatus Network 1 DNS Domain Name

Shows the domain name.

Value space of the result returned:

<String>

Example:

```
xStatus Network DNS Domain Name
*s Network 1 DNS Domain Name: "company.com"
** end
```

xStatus Network 1 DNS Server [n] Address

Shows the IP address of the DNS server.

Value space of the result returned:

<String>

Example:

```
xStatus Network DNS Server 1 Address
*s Network 1 DNS Server 1 Address: "192.0.2.60"
** end
```

NetworkServices status

xStatus NetworkServices

Shows the top level overview of the network services 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: "12.104.193.12 64.104.222.16 144.254.15.121"
** end
```

xStatus NetworkServices NTP CurrentAddress

Returns the address of the NTP server that is currently in use.

Value space of the result returned:

<String>

Example:

```
xStatus NetworkServices NTP CurrentAddress
*s NetworkServices NTP CurrentAddress: "123.254.15.121"
** end
```

xStatus NetworkServices NTP Status

Returns the status of the endpoints synchronizing with the NTP server.
Unknown: State of the synchronization is unknown.
Synced: The system is in sync with the NTP server
Discarded: The NTP result has been discarded.

Value space of the result returned:

<Unknown/Synced/Discarded>

Example:

```
xStatus NetworkServices NTP Status
*s NetworkServices NTP Status: Synced
** end
```

Peripherals status

xStatus Peripherals ConnectedDevice [n] HardwareInfo

Shows hardware information about connected device.

Value space of the result returned:

<String>

Example:

```
xStatus Peripherals ConnectedDevice 1007 HardwareInfo
*s Peripherals ConnectedDevice 1007 HardwareInfo: "1122330-0"
** end
```

xStatus Peripherals ConnectedDevice [n] ID

Shows the MAC-address of the connected device.

Value space of the result returned:

<String>

Example:

```
xStatus Peripherals ConnectedDevice 1007 ID
*s Peripherals ConnectedDevice 1007 ID: "00:10:20:20:be:21"
** end
```

xStatus Peripherals ConnectedDevice [n] Name

Shows the product name of connected device.

Value space of the result returned:

<String>

Example:

```
xStatus Peripherals ConnectedDevice 1007 Name
*s Peripherals ConnectedDevice 1007 Name: "Cisco TelePresence Touch"
** end
```

xStatus Peripherals ConnectedDevice [n] SoftwareInfo

Shows information of the software version running on the connected device.

Value space of the result returned:

<String>

Example:

```
xStatus Peripherals ConnectedDevice 1007 SoftwareInfo
*s Peripherals ConnectedDevice 1007 SoftwareInfo: "TI7.2.0"
** end
```

xStatus Peripherals ConnectedDevice [n] Status

Shows peripheral devices that are currently connected to the endpoint.

Note: Does not apply for the Cisco PrecisionHD cameras.

Value space of the result returned:

<Connected/ResponseTimedOut>

Example:

```
xStatus Peripherals ConnectedDevice 1001 Status
*s Peripherals ConnectedDevice 1001 Status: Connected
** end
```

xStatus Peripherals ConnectedDevice [n] Type

Shows the peripheral types that are connected to the endpoint.

Note: The value space Camera only shows Precision 60 cameras.

Value space of the result returned:

<BluetoothHeadset/Byod/Camera/ControlSystem/ISDNLink/Other/SpeakerTrack/TouchPanel>

Example:

```
xStatus Peripherals ConnectedDevice 1001 Type
*s Peripherals ConnectedDevice 1001 Type: TouchPanel
** end
```

xStatus Peripherals ConnectedDevice [n] UpgradeStatus

Shows the status of the previous software upgrade on the currently connected peripherals.

Value space of the result returned:

<Downloading/Failed/Installing/None/Succeeded>

Example:

```
xStatus Peripherals ConnectedDevice 1001 UpgradeStatus
*s Peripherals ConnectedDevice 1001 UpgradeStatus: None
** 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.

ConfigError: An error occurred during configuration.

Value space of the result returned:

    <Failed/AuthenticationFailed/Provisioned/Idle/NeedConfig/ConfigError>

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 NextRetry
If provisioning has failed, returns the date and time for the next automatic provisioning attempt.

Value space of the result returned:

    <String>

Example:

```
xStatus Provisioning NextRetry
*s Provisioning NextRetry: "2014-04-30T14:02:22Z"
** end
```

xStatus Provisioning Software PreviousUpgrade Changed
Shows the date and time for the previous software upgrade.

Value space of the result returned:

    <String>

Example:

```
xStatus Provisioning Software PreviousUpgrade Changed
*s Provisioning Software PreviousUpgrade Changed: "2014-07-06T09:32:39Z"
** end
```

xStatus Provisioning Software PreviousUpgrade Status
Shows the status of the previous software upgrade.

Value space of the result returned:

    <None/InProgress/Failed/InstallationFailed/Succeeded>

Example:

```
xStatus Provisioning Software PreviousUpgrade Status
*s Provisioning Software PreviousUpgrade Status: None
** end
```

xStatus Provisioning Software PreviousUpgrade Message

Shows the system message for the previous software upgrade.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning Software PreviousUpgrade Message
*s Provisioning Software PreviousUpgrade Message: ""
** end
```

xStatus Provisioning Software PreviousUpgrade VersionId

Shows the version ID of the software uploaded and installed in the previous installation.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning Software PreviousUpgrade VersionId
*s Provisioning Software PreviousUpgrade VersionId: "s52000tc7_2_0.pkg"
** end
```

xStatus Provisioning Software PreviousUpgrade URL

Shows the URL that the previously installed software version was uploaded and installed from.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning Software PreviousUpgrade URL
*s Provisioning Software PreviousUpgrade URL: "http:// 192.0.2.0.6970/
s52000tc7_2_0.pkg"
** 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/FormingHierarchy/Downloading/DownloadPaused/DownloadDone/
Seeding/AboutToInstallUpgrade/Postponed/PeripheralsReady/UpgradingPeripherals/Installing/
InstallingPeripherals>

Example:

```
xStatus Provisioning Software UpgradeStatus Phase
*s Provisioning Software UpgradeStatus Phase: None
** end
```

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

199

Copyright © 2015 Cisco Systems, Inc. All rights reserved.

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.

CAPFNullClnt: Invalid client.

CAPFNullClntcontext: 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/CAPFNullClnt/CAPFNullClntcontext/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 UserId

Returns the current UCM user id for this device. When the Extension Mobility service is in use, this command returns the signed in UserId.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning CUCM UserId
*s Provisioning CUCM UserId: ""
** 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
```

xStatus Provisioning CUCM ITL State

Shows whether the Identity Trust List (ITL) file is installed or not.

Value space of the result returned:

<NotInstalled/Installed>

Example:

```
xStatus Provisioning CUCM ITL State
*s Provisioning CUCM ITL State: Installed
** end
```

xStatus Provisioning CUCM ExtensionMobility Enabled

Indicates if Extension Mobility is enabled for the UCM registered device.

Value space of the result returned:

<True/False>

Example:

```
xStatus Provisioning CUCM ExtensionMobility Enabled
*s Provisioning CUCM ExtensionMobility Enabled: False
** end
```

xStatus Provisioning CUCM ExtensionMobility LoggedIn

Indicates whether you are logged in to Extension Mobility or not.

Value space of the result returned:

<True/False>

Example:

```
xStatus Provisioning CUCM ExtensionMobility LoggedIn
*s Provisioning CUCM ExtensionMobility LoggedIn: False
** end
```

xStatus Provisioning CUCM ExtensionMobility LastLoggedInUserId

Returns the user id last logged in to the system. Only in use if UCM Extension Mobility is enabled, and UCM is configured to remember the last logged in user on an endpoint.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning CUCM ExtensionMobility LastLoggedInUserId
*s Provisioning CUCM ExtensionMobility LastLoggedInUserId: "User 1"
** end
```

xStatus Provisioning CUCM TVS Proxy [n] Server

Returns the address of the TVS server in form of a hostname, FQDN or IPv4 address.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning CUCM TVS Proxy 1 Server
*s Provisioning CUCM TVS Proxy 1 Server: "192.0.2.0"
** end
```

xStatus Provisioning CUCM TVS Proxy [n] IPv6Address

Returns the address of the TVS server in form of a hostname, FQDN or IPv6 address.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning CUCM TVS Proxy 1 IPv6Address
*s Provisioning CUCM TVS Proxy 1 IPv6Address: ""
** end
```

xStatus Provisioning CUCM TVS Proxy [n] Port

Returns the number of the port the TVS is running on.

Value space of the result returned:

<Integer>

Example:

```
xStatus Provisioning CUCM TVS Proxy 1 Port
*s Provisioning CUCM TVS Proxy 1 Port: 2445
** end
```

xStatus Provisioning CUCM TVS Proxy [n] Priority

Returns the Priority of the TVS proxy in use. Lowest priority proxy is preferred.

Value space of the result returned:

<Integer>

Example:

```
xStatus Provisioning CUCM TVS Proxy 1 Priority
*s Provisioning CUCM TVS Proxy 1 Priority: 0
** end
```

xStatus Provisioning CUCM Phonebook URL

Shows the current URL used for phonebook when the system is using CUCM or CUCM via Expressway provisioning. The URL gets updated in failover scenarios.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning CUCM Phonebook URL
*s Provisioning CUCM Phonebook URL: "https://company.cisco.com:8443/cucm-uds/users"
** end
```

Security status

xStatus Security

Shows the top level overview of the 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
```

xStatus Security Persistency Configurations

Shows whether the systems all configurations are set to persistent or non-persistent mode. Persistent is the default mode.

Value space of the result returned:

<NonPersistent/Persistent>

Example:

```
xStatus Security Persistency Configurations
*s Security Persistency Configurations: Persistent
** end
```

xStatus Security Persistency CallHistory

Shows whether call history logging is set to persistent or non-persistent mode. Persistent is the default mode.

Value space of the result returned:

<NonPersistent/Persistent>

Example:

```
xStatus Security Persistency CallHistory
*s Security Persistency CallHistory: Persistent
** end
```

xStatus Security Persistency InternalLogging

Shows whether internal logging is set to persistent or non-persistent mode. Persistent is the default mode.

Value space of the result returned:

<NonPersistent/Persistent>

Example:

```
xStatus Security Persistency InternalLogging
*s Security Persistency InternalLogging: Persistent
** end
```

xStatus Security Persistency LocalPhonebook

Shows whether local phone book is set to persistent or non-persistent mode. Persistent is the default mode.

Value space of the result returned:

<NonPersistent/Persistent>

Example:

```
xStatus Security Persistency LocalPhonebook
*s Security Persistency LocalPhonebook: Persistent
** end
```

xStatus Security Persistency DHCP

Shows whether DHCP logging is set to persistent or non-persistent mode. Persistent is the default mode.

Value space of the result returned:

<NonPersistent/Persistent>

Example:

```
xStatus Security Persistency DHCP
*s Security Persistency DHCP: Persistent
** end
```

xStatus Security Audit Server Port

Returns information on which syslog server port the audit logs are configured to be sent to.

Value space of the result returned:

<Integer>

Example:

```
xStatus Security Audit Server Port
*s Security Audit Server Port: 514
** end
```

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

205

Copyright © 2015 Cisco Systems, Inc. All rights reserved.

SIP status

xStatus SIP
Shows the top level overview of the SIP status.

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.
AuthenticationFailed: Wrong user name or password.

Value space of the result returned:
<Active/DNSFailed/Off/Timeout/UnableTCP/UnableTLS/Unknown/AuthenticationFailed>

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/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 [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 [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 [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 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: "s52000"
** 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: 4
** 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 Encryption

Shows if the system has the option key installed that supports the encryption functionality.

Value space of the result returned:

    <String>

Example:

```
xStatus SystemUnit Software OptionKeys Encryption
*s SystemUnit Software OptionKeys Encryption: "true"
** end
```

xStatus SystemUnit Software OptionKeys RemoteMonitoring

Shows if the system has the option key installed that supports the remote monitoring functionality. Added in TC7.3.3.

Value space of the result returned:

    <String>

Example:

```
xStatus SystemUnit Software OptionKeys RemoteMonitoring
*s SystemUnit Software OptionKeys RemoteMonitoring: "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: "FTT123456A1"
** end
```

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

210

xStatus SystemUnit Hardware Module CompatibilityLevel

The Cisco TelePresence endpoints are using a NAND flash memory for general storage. The endpoints running TC software are manufactured with a new version of the flash memory. Endpoints having the new version of the flash memory must run software TC4.2.1 or later. Some previous TC software versions are updated to support the new flash memory. Please check the release note to find if the new version of the flash memory is supported. If your endpoint does not have the CompatibilityLevel command it does not have the new flash memory installed.

The result returned when running the command is either 0 or 1:

0 = The system does not have the new flash memory installed.

1 = The system has the new flash memory installed. If downgraded, it can only be downgraded to previous TC software versions having support for the new version of the flash memory.

Value space of the result returned:

<"1"/"0">

Example:

```
xStatus SystemUnit Hardware Module CompatibilityLevel
*s SystemUnit Hardware Module CompatibilityLevel: 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: "F1234567"
** 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 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 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 Hardware UDI

Shows the unique device identifier (UDI) for the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Hardware UDI
*s SystemUnit Hardware UDI: "CTS-SX80CODEC V01 FTT123456A1"
** 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 State Subsystem Application

Shows the status of the sub system application.

Initialized: The sub-system application is initialized.

Initializing: The sub-system application is initializing.

Value space of the result returned:

<Initialized/Initializing>

Example:

```
xStatus SystemUnit State Subsystem Application

*s SystemUnit State Subsystem Application: Initialized
** 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 [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 [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 [n] Type

Returns information on the results of the latest diagnostics on the system.

Value space of the result returned:

<CUCMVendorConfigurationFile/CallProtocolDualStackConfig/
CallProtocolIPStackPlatformCompatibility/CallProtocolVcsProvisioningCompatibility/
CameraPairing/CameraSoftwareVersion/CameraStatus/CamerasDetected/
DefaultCallProtocolRegistered/EthernetDuplexMatches/H320GatewayStatus/
H323GatekeeperStatus/HasValidReleaseKey/IPv4Assignment/IPv6Assignment/IPv6Mtu/
ISDNLinkCompatibility/ISDNLinkIpStack/NTPStatus/NetSpeedAutoNegotiated/OSDVideoOutput/
ProvisioningStatus/SIPEncryption/SIPListenPortAndOutboundMode/SIPProfileRegistration/
SIPProfileType/SelectedVideoInputSourceConnected/SipIceAndAnatConflict/
TLSVerifyRequiredCerts/TouchPanelConnection/TurnBandwidth/UdpPortRangeViolation/
ValidPasswords/VideoFromInternalCamera/VideoInputStability/TemperatureCheck>

Example:

```
xStatus SystemUnit Diagnostics Message type
*s SystemUnit Diagnostics Message 1 Type: InvalidAdminPassword
** end
```

xStatus SystemUnit Diagnostics Message [n] 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 [n] 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 [n] 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
Shows the top level overview of the time status.

xStatus Time ZoneOlson
Shows the current time zone in Olson format.

Value space of the result returned:
<String>

Example:
```
xStatus Time ZoneOlson
*s Time ZoneOlson: Europe/Berlin
** end
```

xStatus Time SystemTime
Returns the date and time set on the system.

Value space of the result returned:
<String>

Example:
```
xStatus Time SystemTime
*s Time SystemTime: "2014-04-25T10:04:03Z"
** end
```

UserInterface status

xStatus UserInterface OSD Output
Indicates which monitor is used for on-screen display output.

Value space of the result returned:
<Integer>

Example:
```
xStatus UserInterface OSD Output
*s UserInterface OSD Output: 1
** end
```

xStatus UserInterface OSD Mode
Returns the status of the endpoint's on screen display.

Full: The on screen display is set as the primary user interface and all menu features are available.

Indicators: On screen display is not set as the primary user interface. Only indicators and messages are available.

Value space of the result returned:
<Full/Indicators>

Example:
```
xStatus UserInterface OSD Mode
*s UserInterface OSD Mode: Indicators
** end
```

Video status

xStatus Video Input

Shows the top level overview of the video input status.

xStatus Video Monitors

Returns the monitor layout mode.

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.

Quadruple: The layout is distributed on four monitors, so that each remote participant and the presentation will be shown on separate monitors.

Value space of the result returned:

<Single/Dual/DualPresentationOnly/Triple/Quadruple>

Example:

```
xStatus Video Monitors
*s Video Monitors: Single
** end
```

xStatus Video Input Connector [n] Connected

Shows whether is something connected to the specified connector. Not all connections can be detected.

Value space of the result returned:

<False/True/Unknown>

Example:

```
xStatus Video Input Connector 1 Connected
*s Video Input Connector 1 Connected: True
** end
```

xStatus Video Input Connector [n] SignalState

Shows the signal state for the specified 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:

<OK/Unknown/Unsupported>

Example:

```
xStatus Video Input Connector 1 SignalState
*s Video Input Connector 1 SignalState: OK
** end
```

xStatus Video Input Connector [n] SourceId

Shows the identifer of the input source that the connector is associated with.

Value space of the result returned:

<Integer>

Example:

```
xStatus Video Input Connector 1 SourceId
*s Video Input Connector 1 SourceId: 1
** end
```

xStatus Video Input Connector [n] Type

Shows which connector type it is.

Value space of the result returned:

<Composite/DVI/HDMI/Unknown/YC>

Example:

```
xStatus Video Input Connector 1 Type
*s Video Input Connector 1 Type: HDMI
** end
```

xStatus Video Input LastConnectedSource

Shows the last connected video input source.

Value space of the result returned:

<1..3>

Example:

```
xStatus Video Input LastConnectedSource
*s Video Input LastConnectedSource: 0
** end
```

xStatus Video Input MainVideoSource

Returns the local video input currently used as the main source. The main video source is set with the xConfiguration Video Input VideoMainSource command.

Value space of the result returned:

<Integer>

Example:

```
xStatus Video Input MainVideoSource
*s Video Input MainVideoSource: 1
** end
```

xStatus Video Input Source [n] ConnectorId

Shows the identifier of the connector that is associated with the input source.

Value space of the result returned:

<1..5>

Example:

```
xStatus Video Input Source 1 ConnectorId
*s Video Input Source 1 ConnectorId: 1
** end
```

xStatus Video Input Source [n] 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 [n] 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 [n] 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 [n] 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 [n] Resolution FormatStatus

Shows the resolution format status for the video input source.

Value space of the result returned:

<Ok/OutOfRange/NotFound/Interlaced/Error/Unknown>

Example:

```
xStatus Video Input Source 1 Resolution FormatStatus
*s Video Input Source 1 Resolution FormatStatus: Ok
** end
```

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

217

xStatus Video Input Source [n] MediaChannelId

For internal use only.

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 Output

Shows the top level overview of the video output status.

xStatus Video Output Connector [n] Connected

Describes whether a device (for example a display) is connected to the output connector or not. 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 device is connected to the video output connector.

False: Nothing is connected to the video output connector.

Value space of the result returned:

<True/False>

Example:

```
xStatus Video Output Connector 1 Connected

*s Video Output Connector 1 Connected: True

** end
```

xStatus Video Output Connector [n] MonitorRole

Describes which video stream is shown on the device that is connected to the video output connector.

First/Second/Third: The role of the monitor in a multimonitor setup. In a singlemonitor setup, there is no difference between First, Second and Third.

PresentationOnly: Shows presentation video stream if active.

Recorder: Shows all participants, including the local main video. If active, shows also the presentation.

Value space of the result returned:

<First/Second/Third/PresentationOnly/Recorder>

Example:

```
xStatus Video Output Connector 1 MonitorRole

*s Video Output Connector 1 MonitorRole: First

** end
```

xStatus Video Output Connector [n] Resolution Height

Shows the resolution height (in pixels) for the video output connector.

Value space of the result returned:

<120..3000>

Example:

```
xStatus Video Output Connector 1 Resolution Height

*s Video Output Connector 1 Resolution Height: 1080

** en
```

xStatus Video Output Connector [n] Resolution Width

Shows the resolution width (in pixels) for the video output connector.

Value space of the result returned:

<176..4000>

Example:

```
xStatus Video Output Connector 1 Resolution Width

*s Video Output Connector 1 Resolution Width: 1920

** end
```

xStatus Video Output Connector [n] Resolution RefreshRate

Shows the resolution refresh rate (Hz) for the video output connector.

Value space of the result returned:

<1..300>

Example:

xStatus Video Output Connector 1 Resolution RefreshRate
*s Video Output Connector 1 Resolution RefreshRate: 60
** end

xStatus Video Output Connector [n] Type

Shows the type of connector.
HDMI: It is an HDMI connector.
DVI: It is an DVI connector.

Value space of the result returned:

<HDMI/DVI>

Example:

xStatus Video Output Connector 1 Type
*s Video Output Connector 1 Type: HDMI
** end

xStatus Video Output DVI ConnectedDevice Name

Shows the name of the monitor connected to the DVI port as defined in the monitors EDID.

Value space of the result returned:

<String>

Example:

xStatus Video Output DVI ConnectedDevice Name
*s Video Output DVI 2 ConnectedDevice Name: ""
*s Video Output DVI 4 ConnectedDevice Name: ""
** end

xStatus Video Output DVI ConnectedDevice PreferredFormat

Shows the preferred input format of the monitor connected to the DVI port as defined in the monitors EDID. This is not necessarily the format the codec is sending out.

Value space of the result returned:

<String>

Example:

xStatus Video Output DVI ConnectedDevice PreferredFormat
*s Video Output DVI 2 ConnectedDevice PreferredFormat: ""
** end

xStatus Video Output HDMI ConnectedDevice Name

Shows the name of the monitor connected to the HDMI port as defined in the monitors EDID.

Value space of the result returned:

<String>

Example:

xStatus Video Output HDMI ConnectedDevice Name
*s Video Output HDMI 1 ConnectedDevice Name: ""
*s Video Output HDMI 3 ConnectedDevice Name: ""
** end

xStatus Video Output HDMI ConnectedDevice PreferredFormat

Shows the preferred input format of the monitor connected to the HDMI port as defined in the monitors EDID. This is not necessarily the format the codec is sending out.

Value space of the result returned:

<String>

Example:

xStatus Video Output HDMI ConnectedDevice PreferredFormat
*s Video Output HDMI 1 ConnectedDevice PreferredFormat: ""
** end

xStatus Video Output HDMI ConnectedDevice CEC DeviceType

Shows the type of CEC enabled device connected to the HDMI output the codec has detected.

This information is only available when the device connected to the HDMI output has the CEC feature configured on and the codec has the configuration xConfiguration Video Output HDMI [n] CEC Mode set to on.

Value space of the result returned:

<Unknown/TV/Reserved/Recorder/Tuner/Playback/Audio>

Example:

```
xStatus Video Output HDMI ConnectedDevice CEC DeviceType
*s Video Output HDMI 1 ConnectedDevice CEC 1 DeviceType: TV
** end
```

xStatus Video Output HDMI ConnectedDevice CEC PowerControl

Shows whether the codec is controlling the CEC enabled device connected to the HDMI output.

This information is only available when the device connected to the HDMI output has the CEC feature configured on and the codec has the configuration xConfiguration Video Output HDMI [n] CEC Mode set to on.

Value space of the result returned:

<Unknown/Ok/In progress/Failed to power on/Failed to standby>

Example:

```
xStatus Video Output HDMI ConnectedDevice CEC PowerControl
*s Video Output HDMI 1 ConnectedDevice CEC 1 PowerControl: Ok
** end
```

xStatus Video Output HDMI ConnectedDevice CEC PowerStatus

Shows the state of the CEC enabled devise connected to the HDMI output.

This information is only available when the device connected to the HDMI output has the CEC feature configured on and the codec has the configuration xConfiguration Video Output HDMI [n] CEC Mode set to on.

Value space of the result returned:

<Unknown/Ok/In progress/Failed to power on/Failed to standby>

Example:

```
xStatus Video Output HDMI ConnectedDevice CEC PowerStatus
*s Video Output HDMI 1 ConnectedDevice CEC 1 PowerStatus: Standby
** 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 [n] Output [n] 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 [n] Output [n] 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 [n] Output [n] 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 [n] Output [n] Frame [n] 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 [n] Output [n] Frame [n] 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 [n] Output [n] Frame [n] 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 [n] Output [n] Frame [n] 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 [n] Output [n] Frame [n] 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 [n] Output [n] Frame [n] MediaChannelId

For internal use only.

Value space of the result returned:

<Integer>

Example:

```
xStatus Video Layout Site Output  Frame MediaChannelId
*s Video Layout Site 1 Output 1 Frame 1 MediaChannelId: 157
*s Video Layout Site 1 Output 2 Frame 1 MediaChannelId: 157
*s Video Layout Site 1 Output 3 Frame 1 MediaChannelId: 157
*s Video Layout Site 1 Output 4 Frame 1 MediaChannelId: 157
*s Video Layout Site 1 Output 5 Frame 1 MediaChannelId: 157
** end
```

xStatus Video Layout Site [n] Output [n] Frame [n] 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 [n] Output [n] Frame [n] 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:

<Integer>

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 VideoSourceId
*s Video Layout Site 1 Output 1 Frame 1 VideoSourceId: 1
** end
```

xStatus Video Layout Site [n] Output [n] Frame [n] InputNumber

Shows the layout input number.

Value space of the result returned:

<1..4>

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 InputNumber
*s Video Layout Site 1 Output 1 Frame 1 InputNumber: 1
** end
```

xStatus Video Layout Site [n] Output [n] Frame [n] 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/wallpa-
per.png"
** end
```

xStatus Video Layout Site [n] Output [n] Frame [n] 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 Layout Site [n] Output [n] Frame [n] FrameContainerId

For internal use only.

Value space of the result returned:

<Integer>

xStatus Video Layout Site [n] Output [n] FrameContainer [n] PositionX

For internal use only.

Value space of the result returned:

<Integer>

xStatus Video Layout Site [n] Output [n] FrameContainer [n] PositionY

For internal use only.

Value space of the result returned:

<Integer>

xStatus Video Layout Site [n] Output [n] FrameContainer [n] Width

For internal use only.

Value space of the result returned:

<Integer>

xStatus Video Layout Site [n] Output [n] FrameContainer [n] Height

For internal use only.

Value space of the result returned:

<Integer>

xStatus Video Layout Site [n] Output [n] FrameContainer [n] Content

For internal use only.

Value space of the result returned:

<Integer>

xStatus Video Layout Prediction Site [n] Family[n] OutputRole [n] Frame[n] ParticipantId

For internal use only.

Value space of the result returned:

<String>

xStatus Video Layout Prediction Site [n] Family [n] OutputRole [n] Frame[n]
VideoSourceInstance

For internal use only.

Value space of the result returned:

<Integer>

xStatus Video Layout Prediction Site [n] StatusIndex Family [n] Hidden

For internal use only.

Value space of the result returned:

<False/True>

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
```

xStatus Video OSD Output

This status is replaced with xStatus UserInterface OSD Output.

xStatus Video OSD Mode

Deprecated; xStatus Video OSD Mode.

Value space of the result returned:

<Indicators/Full>

Example:

```
xStatus Video OSD Mode
*s Video OSD Mode: Indicators
** 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.

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

225

Chapter 6

Appendices

About startup scripts

You can add one or more startup scripts to the codec. A startup script contains commands (xCommand) and configurations (xConfiguration) that will be executed as part of the start up procedure every time the codec boots. A few commands and configurations cannot be placed in a startup script, e.g. xCommand Boot.

Use the web interface of the codec to create and manage startup scripts.

Read more about the web interface and startup scripts in the Administrator guide for your product.

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
idefixversion
touchpanelversion
license
network
ntp
pairing
passwd
pki
rootsettings
securitysettings
securitystatus
selectsw
sudo
whoami
```

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 idefixversion

Returns the software version of a Touch controller that is connected to the codec.

systemtools license list

Lists all the licenses for the codec.

systemtools license show <name>

Shows the content of a license file, defined 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

Shows the systems IP addresses.

systemtools network ifconfig

Network debug command.

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 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 simulta-neous 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.

systemtools selectsw

Select which of the available software images to use. Changing the software image will restart the codec.

No argument given: Lists the version of the available software images, and shows which one is active.

image name: Swap to the software with this name.

systemtools sudo on

Changes the user role of the current session to the role of the specified user

username(r): The name of the sudo user.

password(r): The password for the sudo user.

systemtools sudo off

Reverts back to the user role of the signed in user.

systemtools whoami

Lists the name and id of the signed in user, and the user roles held by this user.

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

229

About disconnect cause types

The following parameters are logged when a call is discon-
nected. The disconnect cause types are used in disconnect
events (xEvent).

CauseValue	Proprietary. We recomend the use of CauseType and CauseCode.
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:

```
 xEvent DisconnectEvent

*e CallDisconnect CauseValue: 1
    CauseType: "LocalDisconnect"
    CauseString: ""
    OrigCallDirection: "outgoing"
    RemoteURI: "firstname.lastname@company.com"
    CallId: 89
    CauseCode: 0
    CauseOrigin: SIP
** end
```

Intellectual property rights

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies are considered un-Controlled copies and the original on-line version should be referred to for latest version.

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

TANDBERG is now a part of Cisco. TANDBERG® is a registered trademark belonging to Tandberg ASA.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Cisco contacts

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

D15107.05 Codec SX80 API Reference Guide TC7.3, OCTOBER 2015.

231

Copyright © 2015 Cisco Systems, Inc. All rights reserved.