

Collaboration Endpoint Software 8.3  
JANUARY 2017



# Application Programming Interface (API) Reference Guide

Cisco TelePresence DX70 and DX80

## 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/dx-docs](http://www.cisco.com/go/dx-docs)

## Table of Contents

<b>Introduction.....</b>	<b>3</b>	Using HTTP.....	20
About this guide.....	4	HTTP XMLAPI Authentication.....	20
Software migration to CE software.....	4	HTTP XMLAPI Session Authentication.....	20
Downloading the user documentation.....	4	Getting status and configurations.....	21
What's new in CE8.3.....	5	Sending commands and configurations.....	21
		Using HTTP POST.....	21
<b>About the API.....</b>	<b>10</b>	Feedback from codec over HTTP.....	22
API fundamentals.....	11	Registering for feedback.....	22
Connecting to the API.....	11	Translating from terminal mode to XML.....	23
Password.....	11	Dos and don'ts.....	24
SSH.....	11	<b>xConfiguration commands.....</b>	<b>25</b>
Telnet.....	11	Description of the xConfiguration commands.....	26
HTTP/HTTPS.....	11	<b>xCommand commands.....</b>	<b>72</b>
Serial connection over micro USB.....	12	Description of the xCommand commands.....	73
API output.....	13	<b>xStatus commands.....</b>	<b>110</b>
The output modes.....	13	Description of the xStatus commands.....	111
Using the command line.....	14	<b>Appendices.....</b>	<b>162</b>
API commands.....	14	About startup scripts.....	163
Other commands.....	14	The SystemTools commands.....	164
Command line shortcuts.....	15	About disconnect cause types.....	166
Searching.....	15	User documentation on the Cisco web site.....	167
Value types and formats.....	15		
Commands.....	16		
Configurations.....	16		
Status.....	16		
Events.....	17		
Feedback mechanism.....	18		
Feedback expressions.....	18		
Terminal connections.....	19		



## Chapter 1

# Introduction

## About this guide

This guide introduces you to the Application Programmer Interface (API) for the Collaboration Endpoint software, and serves as a reference guide for the command line commands for the following product:

- Cisco TelePresence DX70 and DX80

**NOTE:** This guide doesn't apply to Cisco Spark registered devices. To find more information on Cisco Spark room devices, visit: [▶ https://help.webex.com/](https://help.webex.com/)

## Software migration to CE software

From Collaboration Software version 8.2 (CE8.2) all DX80 and DX70 units can run *CE software*, which is the same software that runs on the Cisco TelePresence SX and MX Series.

Cisco DX80 and Cisco DX70 were originally shipped with *Android based software*. Before migrating to *CE software*, it is important to carefully consider the conversion requirements and the functionality changes compared to *Android based software*; otherwise migration can leave you with a non-functional deployment that requires you to convert back.

Refer to the software release notes, and the Administrator Guide for further details.

## Downloading the user documentation

You can download the user documentation from the Cisco web site, go to: [▶ http://www.cisco.com/go/dx-docs](http://www.cisco.com/go/dx-docs)

Guidelines for how to find the documentation on the Cisco web site are included in the [▶ User documentation on the Cisco web site](#) appendix.

## What's new in CE8.3

This section provides an overview of the new and changed system settings, and the new features and improvements in the Cisco Collaboration Endpoint software version 8.3 (CE8.3) compared to CE8.2.

For more details, we recommend reading the Software release notes:

► <http://www.cisco.com/c/en/us/support/collaboration-endpoints/desktop-collaboration-experience-dx600-series/products-release-notes-list.html>

### New features and improvements

#### Wi-Fi

DX70 and DX80 video systems that are shipped with a built-in Wi-Fi adapter, can connect to the network either via Ethernet or Wi-Fi.

The video system supports the following standards for Wi-Fi:

- IEEE 802.11a
- IEEE 802.11b
- IEEE 802.11g
- IEEE 802.11n

In addition to open networks (not secured), the following security protocols are supported:

- WPA-PSK (AES)
- WPA2-PSK (AES)

The video system cannot connect to both Ethernet and Wi-Fi at the same time. If connected, Ethernet takes precedence over Wi-Fi, and you must disconnect the Ethernet cable in order to use Wi-Fi.

#### Intelligent Proximity for content sharing

In alignment with the other video systems running CE software, DX70 and DX80 now support Cisco Proximity. Cisco Proximity allows you to automatically pair your smartphone, tablet, or laptop with the video system when the device comes within range. This feature is disabled by default, but you can enable it from the web interface or xAPI.

DX70 and DX80 support the same services as other video systems with CE software:

- Content sharing to mobile clients (only when in a call due to privacy concerns)
- Content sharing from desktop clients
- Basic call control from mobile clients

Cisco recommends only one video system with Proximity enabled per room. More systems may cause unstable pairing conditions. The maximum number of simultaneous Proximity connections is three.

The Cisco Proximity clients for smartphones and tablets (Android and iOS), and laptops (Windows and OS X) can be downloaded from ► <http://proximity.cisco.com>. Clients for smartphones and tablets are also available through Google Play (Android) and Apple App Store (iOS).

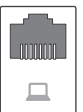
#### More Cisco UCM features

Multiple Cisco UCM features, which are already supported by the other video systems that are running CE software, are enabled:

- Voicemail
- Shared lines
- Extension mobility
- Forward all calls
- Consultative add and transfer
  - Allows you to consult with the person you intend to add or transfer to before completing the action
- Multiple calls on hold, and merge calls

#### IEEE 802.1x authentication for devices connected to the computer network port

IEEE 802.1x authentication is enabled on the *computer network* port of DX70 and DX80. This means that devices connected to this port can authenticate using this protocol.



### In-Room Controls

With In-Room Control you can customize the video system's user interface to allow control of peripherals in your room, for example lights and blinds. You get a consistent user experience when the video system and other peripherals in the room are controlled from the same device.

You may experience low resolution graphics in the first release. Also see the Release Note.

For more information about setting up the In-Room Control feature, refer to the user guide: [▶ http://www.cisco.com/go/in-room-control-docs](http://www.cisco.com/go/in-room-control-docs)

### Control of an external video switch with additional input sources

You can customize our touch user interfaces to include input sources that are connected to third-party external video switches. These *external* input sources will appear and behave just as a video source that is connected directly to the video system. You don't have to swap cables or change the physical input source manually.

The number of external input sources you can create depends on how many inputs the video switches allow. Cisco recommends maximum 50 sources.

The video switch must be supported by a third-party control system, for example from Crestron or AMX, which can interact with the video system's xAPI. It is the control system, not the video system, that controls the video switch.

For more information about how to customize the user interface, and how to use the xAPI to set up this feature, refer to the user guide: [▶ http://www.cisco.com/go/in-room-control-docs](http://www.cisco.com/go/in-room-control-docs)

### ActiveControl in CMS 2.1 hosted conferences

ActiveControl is a SIP-only feature that has been supported for a while for video systems in Cisco TelePresence Server hosted conferences.

From CE8.3 ActiveControl is also supported in CMS 2.1 hosted conferences. It is enabled by default on the video system, and can be used as long as it is supported by the infrastructure as well (auto-negotiation).

New ActiveControl functionality:

- When the video system is muted from the server side, the mute indicator is shown on screen and the microphone's mute LED is lit.
- Best effort encryption is used by default.
- An *automatic* layout is introduced as default. Who is shown in the *automatic* layout is based on recent active speakers.
- Recording status indicator shown on screen.
- ActiveControl functionality available from the Call Control page on the web interface, for example participant list, participant mute indicator, and drop participant button.

### Media channel details available in the xAPI

Media channels contains information about audio, video and data for ongoing calls. You can monitor the channel rate, packet loss, jitter, video frame rate, and more from the xAPI.

Sign in to the xAPI as an admin user and run `xStatus MediaChannels` to see the media channel details. You can also check the media channel status on the Status page in the web interface.

### HTTP feedback in JSON format

HTTP feedback can be sent in JSON (JavaScript Object Notation) format for third-party feedback interpreters, as an alternative to XML (eXtensible Markup Language).

TMS uses XML by default, and does not support JSON formatted feedback.

### Configurable network speed

You can configure the Ethernet link speed from the web interface or the xAPI. This is in alignment with other video systems running CE software.

You can choose to autonegotiate the link speed (default), or force it to:

- 10 Mbps half-duplex
- 10 Mbps full-duplex
- 100 Mbps half-duplex
- 100 Mbps full-duplex
- 1 Gbps full-duplex
- In general, we recommend not to change from the default value, where the video system negotiates with the network to set the speed automatically.

### Keypad for DTMF tones available on the web interface

We have added a keypad to the Call Control page of the web interface. The keypad is available when in call, and can be used to send DTMF tones to the far end when required.

### New languages

We have added support for EnglishUK and SpanishLatin to the on-screen display and user interfaces.

## xConfiguration changes in CE8.3

**New configurations**

Audio Microphones Mute Enabled  
Network [1] Speed  
Peripherals Profile ControlSystems  
Proximity Mode  
Proximity Services CallControl  
Proximity Services ContentShare FromClients  
Proximity Services ContentShare ToClients

**Configurations that are removed**

SIP Ice OfferTcpCandidates

**Configurations that are modified**

H323 Encryption KeySize  
**OLD:** Default value: Max1024bit  
**NEW:** Default value: Min1024bit

Network [1] \*  
**OLD:** User role: Admin  
**NEW:** User role: Admin, User

UserInterface/Language  
**Change:** EnglishUK and SpanishLatin added to value space

UserInterface OSD Output  
**OLD:** 1 / 2 / 3 / Auto  
Default value: 1  
**NEW:** Auto  
Default value: Auto

Video Monitors  
**OLD:** Auto / Dual / DualPresentationOnly / Single  
**NEW:** Single

Video Selfview Default OnMonitorRole  
**OLD:** Current / First / Fourth / Second / Third  
Default value: Current  
**NEW:** First  
Default value: First

### xCommand changes in CE8.3

#### New commands

- Audio Diagnostics MeasureDelay
- Call Forward
- Provisioning CUCM ExtensionMobility Login
- Provisioning CUCM ExtensionMobility Logout
- UserInterface Extensions Clear
- UserInterface Extensions List
- UserInterface Extensions Widget SetValue
- UserInterface Extensions Widget UnsetValue
- UserInterface Presentation ExternalSource Add
- UserInterface Presentation ExternalSource List
- UserInterface Presentation ExternalSource Remove
- UserInterface Presentation ExternalSource RemoveAll
- UserInterface Presentation ExternalSource Select
- UserInterface Presentation ExternalSource State Set)

#### Commands that are removed

- Video Input Source SetActiveConnector

#### Commands that are modified

- Call Resume
  - OLD: CallId: required
  - NEW: CallId: optional
- HttpFeedback Register
  - OLD: FeedbackSlot: optional
  - NEW: FeedbackSlot: required
  - Format: XML/JASON
- Peripherals List
  - OLD: User role: Admin
  - NEW: User role: Admin, RoomControl

- SystemUnit OptionKey Remove)
  - OLD: Encryption / MultiSite / RemoteMonitoring
  - NEW: RemoteMonitoring
- SystemUnit SoftReset
  - OLD: User role: Admin
  - NEW: User role: Admin, User
- UserInterface Message Alert Clear
  - OLD: User role: Admin, User
  - NEW: User role: Admin, User, RoomControl
- UserInterface Message Alert Display
  - OLD: User role: Admin
  - NEW: User role: Admin, RoomControl
- UserInterface Message Prompt Clear
  - OLD: User role: Admin, User
  - NEW: User role: Admin, User, RoomControl
- UserInterface Message Prompt Display
  - OLD: User role: Admin
  - NEW: User role: Admin, RoomControl
- UserInterface Message Prompt Response
  - OLD: User role: Admin, User
  - NEW: User role: Admin, User, RoomControl
- UserInterface Message TextLine Clear
  - OLD: User role: Admin, User
  - NEW: User role: Admin, User, RoomControl
- UserInterface Message TextLine Display
  - OLD: User role: Admin
  - NEW: User role: Admin, RoomControl
- UserManagement User Passphrase Change
  - OLD: User role: Admin, User
  - NEW: User role: Admin, User, Audit, RoomControl



## xStatus changes in CE8.3

**New statuses**

HttpFeedback [n] Format  
MediaChannels Call [n] Channel [n] Audio Channels  
MediaChannels Call [n] Channel [n] Audio Mute  
MediaChannels Call [n] Channel [n] Audio Protocol  
MediaChannels Call [n] Channel [n] Direction  
MediaChannels Call [n] Channel [n] Encryption  
MediaChannels Call [n] Channel [n] Netstat Bytes  
MediaChannels Call [n] Channel [n] Netstat ChannelRate  
MediaChannels Call [n] Channel [n] Netstat Jitter  
MediaChannels Call [n] Channel [n] Netstat LastIntervalLost  
MediaChannels Call [n] Channel [n] Netstat LastIntervalReceived  
MediaChannels Call [n] Channel [n] Netstat Loss  
MediaChannels Call [n] Channel [n] Netstat MaxJitter  
MediaChannels Call [n] Channel [n] Netstat Packets  
MediaChannels Call [n] Channel [n] ParticipantId  
MediaChannels Call [n] Channel [n] Type  
MediaChannels Call [n] Channel [n] Video ChannelRole  
MediaChannels Call [n] Channel [n] Video FrameRate  
MediaChannels Call [n] Channel [n] Video Protocol  
MediaChannels Call [n] Channel [n] Video ResolutionX  
MediaChannels Call [n] Channel [n] Video ResolutionY  
Provisioning CUCM ExtensionMobility Enabled  
Provisioning CUCM ExtensionMobility LastLoggedInUserId  
Provisioning CUCM ExtensionMobility LoggedIn  
UserInterface Extensions Widget [n] Value  
UserInterface Extensions Widget [n] WidgetId

**Statuses that are removed**

UserInterface OSD Mode

**Statuses that are modified**

Diagnostics Message [n] Type

**Changed:** ControlSystemConnection and AudioECRefDelay added to value space

NetworkServices NTP Status

**OLD:** Discarded / Synced / NotSynced / Unknown

**NEW:** Discarded / Synced / NotSynced / Unknown / Off



## Chapter 2

# About the API

## API fundamentals

This chapter contains a top-level view of the mechanisms supported by the codec API.

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. This chapter also describes how to use the feedback functions that are available for the codec.

The API consists of four major groups:

- Commands
- Configurations
- Status
- Events

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

### Connecting to the API

There are several ways to access the codec API. Regardless of which method you choose, the structure of the API stays the same. Choose the connection method that suits 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 configurations, apart from password, can be set from the System configuration menu on the web interface or from the command line interface. All of the examples are for the command line interface.

### Password

The video system is delivered with a default user account with full credentials. The user name is admin, and initially, no password is set for the default user.

It is mandatory to set a password for the admin user in order to restrict access to system configuration. You should in addition set a password for any other user with similar credentials.

You can set the password on the codec's web interface. Open a web browser and enter the IP address of the video system in the address bar and log in. Click your user name in the upper right corner and choose *Change password* in the drop down menu.

### 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.

```
xConfiguration NetworkServices SSH Mode:
<Off,0n>
```

### 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.

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:
<Off,0n>
```

### HTTP/HTTPS

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

In order to enable or disable the HTTP and HTTPS services, configure the following setting on the codec.

```
xConfiguration NetworkServices HTTP Mode:
<Off, HTTP+HTTPS, HTTPS>
```

### 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 [Integration > Developer API](#). The HTTP POST and GET methods are used to execute commands and get feedback from the codec. This is described in [▶"Using HTTP" on page 20](#) in this document.

## Serial connection over micro USB

You can use the micro USB connector for direct communication with the video system. You may need to install a serial port driver on your computer manually.

Serial connection is allowed by default. It is mainly intended for debugging.

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

## Baud-rate and other serial port parameters

The serial port uses 115200 bps, 8 data bits, no parity and 1 stop bit.

## Login and password

You can choose whether login is required or not for serial connections. For security reasons the password prompting is turned on by default, but it can be turned off if preferred.

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

## 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.
- **JSON:** JSON format is convenient when integrating with web based systems.

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 output 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 (Example 1).

### Bye

Typing the `bye` command closes the command line interface.

```

Example 1:

?
          - User Commands -

help      xcommand      xconfiguration  xevent      xfeedback
xgetxml   xdocument      xpreferences    xtransaction 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 16](#).

### 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 16](#).

### 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 18](#).

### xPreferences

The xPreferences command is used to set preferences for the RS-232, Telnet and SSH sessions. Refer to ["API output" on page 13](#).

### 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` user input is not 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 17](#).

### xGetxml

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

## Other commands

### Systemtools

The systemtools commands are 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 164](#).

### 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).

**NOTE:** 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.

### 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
```

### 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
```

## Value types and formats

The system supports the following value types:

- **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.

### Example:

```
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
```

## 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.

When issuing a `xCommand`, the command consists of one argument and one required parameter. In this document the command usage is described for all `xCommands`, with both required and optional parameters. The optional parameters are in brackets.

### 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 `:'`.

## 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 some of the configurations, 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
```

## 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 (address expression):

```
xStatus <address expression>
```



## 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 dialed. 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 Successful 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 your application and the codec synchronized, you can 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 you have explicitly told the codec to do so. You can subscribe to the feedback by registering feedback expressions. How you register 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 the ["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. Do only subscribe to the feedback 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. The XPath language describes a way to select nodes from an XML/JSON document. The CE software contains three main feedback documents.

Document	API command	Path
Status	<code>xStatus</code>	<code>/Status</code>
Configuration	<code>xConfiguration</code>	<code>/Configuration</code>
Event	<code>xEvent</code>	<code>/Event</code>

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 session. 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.

### HTTP XMLAPI Authentication

Access to the XMLAPI requires the user to authenticate using HTTP Basic Access Authentication as a user with 'ADMIN' role. Unauthenticated requests prompt a 401 HTTP response containing a Basic Access Authentication challenge. How to use HTTP Basic Access Authentication varies according to the HTTP library or tool that you are using.

If your application will be issuing multiple commands through the API, we highly recommend that you use Session Authentication (see below). The standard basic authentication does a full re-authentication per request, which may affect the performance of your application.

### HTTP XMLAPI Session Authentication

Authenticating with your username and password combination for each API request might introduce too much latency for some use-cases. To mitigate this, the API supports a session-based authentication mechanism.

To open a session, issue a POST to <http://<ip-address>/xmlapi/session/begin> with Basic Access Authentication. The response sets a SessionId-cookie that can be used with subsequent requests.

Note that when using API Session Authentication, it is important to explicitly close the session when you are done. Failing to do so may cause the device to run out of sessions, as there are a limited number of concurrent sessions available, and they do not time out automatically.

### 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

#### Example: Starting a HTTP XMLAPI session

##### Request:

```
POST /xmlapi/session/begin HTTP/1.1
Authorization: Basic <Base64 encoded authentication string>
```

##### Response:

```
HTTP/1.1 204 No Content
Server: nginx/1.8.0
Connection: keep-alive
Set-Cookie: SessionId=f08102c8ce5aaf8fba23a7238cc2ef464b990e18bfb7fb048820c0e28955c54; Path=/; HttpOnly
```

This session counts toward the systems concurrent sessions limit.

With an open session, provide the SessionId cookie to following requests. Your tool/library may do this automatically.

#### Example: Using a HTTP XMLAPI session

##### Request:

```
GET /configuration.xml HTTP/1.1
Cookie: SessionId=f08102c8ce5aaf8fba23a7238cc2ef464b990e18bfb7fb048820c0e28955c54
```

##### Response:

```
HTTP/1.1 200 OK
Server: nginx/1.8.0
Content-Type: text/xml; charset=UTF-8
Content-Length: 43549
Connection: keep-alive
<?xml version="1.0"?>
<Configuration product="Cisco Codec"
version="ce8.2.0" apiVersion="4"> ...
</Configuration>
```

To close a session after use, issue a POST to <http://<ip-address>/xmlapi/session/end> with the provided cookie.

#### Example: Closing a HTTP XMLAPI session

##### Request:

```
POST /xmlapi/session/end HTTP/1.1
Cookie: SessionId=f08102c8ce5aaf8fba23a7238cc2ef464b990e18bfb7fb048820c0e28955c54
```

##### Response:

```
HTTP/1.1 204 No Content
Server: nginx/1.8.0
Connection: keep-alive
Set-Cookie: SessionId=; Max-Age=0; Path=/; HttpOnly
```

## 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 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
```

```
<Configuration>
  <SystemUnit>
    <Name>newName</Name>
  </SystemUnit>
</Configuration>
```

#### Example 2: Setting the camera position.

##### Request

```
POST /putxml HTTP/1.1
Content-Type: text/xml
```

```
<Command>
  <Camera>
    <PositionSet command="True">
      <CameraId>1</CameraId>
      <Pan>200</Pan>
      <Tilt>200</Tilt>
    </PositionSet>
  </Camera>
</Command>
```

##### Response

```
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: 91
```

```
<?xml version="1.0"?>
<Command>
<CameraPositionSetResult item="1" status="OK"/>
</Command>
```

## Feedback from codec over HTTP

You can get the device to post http feedback messages (also known as webhooks) on changes to the API state, e.g. statuses, events and configuration updates. The HTTP Post feedback messages will be sent to the specified ServerURL. You can choose between events being posted in either XML or JSON format. You can subscribe to changes on multiple parts of the API by register up to 15 different feedback expressions.

### 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>
  Format: <XML/JSON>
  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 feedback slot to register its expressions.

**ServerUrl:** The URL to the HTTP server where you want the codec to post the HTTP feedback messages to.

**Format:** Set the format for the feedback from the HTTP server to XML or JSON.

**Expression 1-15:** The XPath expression specifies which parts of the Status, Configuration or Event documents are monitored. You can have from 1 to 15 XPath expressions

Register the expressions you want to receive feedback on. See the ["Feedback mechanism" on page 18](#) section for more information about the expression formats.

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

```
POST /putxml HTTP/1.1
Content-Type: text/xml

<Command>
  <HttpFeedback>
    <Register command="True">
      <FeedbackSlot>1</FeedbackSlot>
      <ServerUrl>http://127.0.0.1/myhttppostscripturl</ServerUrl>
      <Format>XML</Format>
      <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>My System Name</SystemName>
    <MACAddress>00:00:de:ad:be:ef</MACAddress>
    <IPAddress>192.168.1.100</IPAddress>
    <ProductType>Cisco Codec</ProductType>
    <ProductID>Cisco Codec DX80</ProductID>
    <SWVersion>CE8.3.0.199465</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>
```

## Dos and don'ts

Here are some best practices when programming the Cisco series API.

### 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.



## Chapter 3

# xConfiguration commands

## Description of the xConfiguration commands

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

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

Go to: <http://www.cisco.com/go/dx-docs>

<b>Audio configuration</b> .....	<b>29</b>	xConfiguration H323 CallSetup Mode.....	34
xConfiguration Audio DefaultVolume .....	29	xConfiguration H323 Encryption KeySize .....	34
xConfiguration Audio Microphones Mute Enabled.....	29	xConfiguration H323 Gatekeeper Address .....	34
xConfiguration Audio SoundsAndAlerts RingTone.....	29	xConfiguration H323 H323Alias E164 .....	35
xConfiguration Audio SoundsAndAlerts RingVolume .....	29	xConfiguration H323 H323Alias ID.....	35
xConfiguration Audio Input MicrophoneMode .....	30	xConfiguration H323 NAT Mode .....	35
		xConfiguration H323 NAT Address .....	36
<b>Conference configuration</b> .....	<b>30</b>	<b>Network configuration</b> .....	<b>36</b>
xConfiguration Conference AutoAnswer Mode .....	30	xConfiguration Network [1] DNS Domain Name.....	36
xConfiguration Conference AutoAnswer Mute .....	30	xConfiguration Network [1] DNS Server [1..3] Address .....	36
xConfiguration Conference AutoAnswer Delay .....	31	xConfiguration Network [1] IEEE8021X Mode.....	37
xConfiguration Conference DefaultCall Rate .....	31	xConfiguration Network [1] IEEE8021X TlsVerify .....	37
xConfiguration Conference DoNotDisturb DefaultTimeout.....	31	xConfiguration Network [1] IEEE8021X UseClientCertificate .....	37
xConfiguration Conference Encryption Mode.....	31	xConfiguration Network [1] IEEE8021X Identity .....	37
xConfiguration Conference FarEndControl Mode.....	32	xConfiguration Network [1] IEEE8021X Password.....	38
xConfiguration Conference MaxReceiveCallRate .....	32	xConfiguration Network [1] IEEE8021X AnonymousIdentity.....	38
xConfiguration Conference MaxTransmitCallRate.....	32	xConfiguration Network [1] IEEE8021X Eap Md5.....	38
xConfiguration Conference MaxTotalReceiveCallRate.....	32	xConfiguration Network [1] IEEE8021X Eap Tls .....	38
xConfiguration Conference MaxTotalTransmitCallRate.....	33	xConfiguration Network [1] IEEE8021X Eap Tls .....	39
		xConfiguration Network [1] IEEE8021X Eap Peap.....	39
<b>H323 configuration</b> .....	<b>33</b>	xConfiguration Network [1] IPStack .....	39
xConfiguration H323 Authentication Mode.....	33	xConfiguration Network [1] IPv4 Assignment.....	39
xConfiguration H323 Authentication LoginName.....	33	xConfiguration Network [1] IPv4 Address.....	40
xConfiguration H323 Authentication Password .....	34		

xConfiguration Network [1] IPv4 Gateway .....	40	xConfiguration NetworkServices SSH Mode .....	49
xConfiguration Network [1] IPv4 SubnetMask .....	40	xConfiguration NetworkServices Telnet Mode .....	50
xConfiguration Network [1] IPv6 Assignment .....	40	xConfiguration NetworkServices WelcomeText.....	50
xConfiguration Network [1] IPv6 Address .....	41	<b>Peripherals configuration .....</b>	<b>50</b>
xConfiguration Network [1] IPv6 Gateway .....	41	xConfiguration Peripherals Profile ControlSystems.....	50
xConfiguration Network [1] IPv6 DHCPOptions.....	41	<b>Phonebook configuration .....</b>	<b>51</b>
xConfiguration Network [1] MTU .....	41	xConfiguration Phonebook Server [1] ID .....	51
xConfiguration Network [1] QoS Mode.....	42	xConfiguration Phonebook Server [1] Type.....	51
xConfiguration Network [1] QoS Diffserv Audio .....	42	xConfiguration Phonebook Server [1] URL.....	51
xConfiguration Network [1] QoS Diffserv Video .....	43	<b>Provisioning configuration .....</b>	<b>52</b>
xConfiguration Network [1] QoS Diffserv Data .....	43	xConfiguration Provisioning Connectivity .....	52
xConfiguration Network [1] QoS Diffserv Signalling .....	44	xConfiguration Provisioning Mode .....	52
xConfiguration Network [1] QoS Diffserv ICMPv6.....	44	xConfiguration Provisioning LoginName .....	53
xConfiguration Network [1] QoS Diffserv NTP .....	45	xConfiguration Provisioning Password.....	53
xConfiguration Network [1] RemoteAccess Allow.....	45	xConfiguration Provisioning HttpMethod .....	53
xConfiguration Network [1] Speed .....	45	xConfiguration Provisioning ExternalManager Address.....	53
xConfiguration Network [1] VLAN Voice Mode .....	46	xConfiguration Provisioning ExternalManager AlternateAddress .....	54
xConfiguration Network [1] VLAN Voice VlanId.....	46	xConfiguration Provisioning ExternalManager Protocol.....	54
<b>NetworkServices configuration .....</b>	<b>46</b>	xConfiguration Provisioning ExternalManager Path .....	54
xConfiguration NetworkServices CDP Mode.....	46	xConfiguration Provisioning ExternalManager Domain .....	54
xConfiguration NetworkServices H323 Mode .....	46	<b>Proximity configuration.....</b>	<b>55</b>
xConfiguration NetworkServices HTTP Mode .....	47	xConfiguration Proximity Mode .....	55
xConfiguration NetworkServices HTTPS VerifyServerCertificate .....	47	xConfiguration Proximity Services CallControl .....	55
xConfiguration NetworkServices HTTPS VerifyClientCertificate.....	47	xConfiguration Proximity Services ContentShare FromClients .....	55
xConfiguration NetworkServices NTP Mode .....	47	xConfiguration Proximity Services ContentShare ToClients.....	55
xConfiguration NetworkServices NTP Server [1..3] Address.....	48	<b>SerialPort configuration.....</b>	<b>56</b>
xConfiguration NetworkServices SIP Mode.....	48	xConfiguration SerialPort Mode .....	56
xConfiguration NetworkServices SNMP Mode.....	48	xConfiguration SerialPort LoginRequired.....	56
xConfiguration NetworkServices SNMP Host [1..3] Address.....	48		
xConfiguration NetworkServices SNMP CommunityName .....	49		
xConfiguration NetworkServices SNMP SystemContact .....	49		
xConfiguration NetworkServices SNMP SystemLocation .....	49		

<b>SIP configuration</b> .....	<b>56</b>	<b>UserManagement configuration</b> .....	<b>64</b>
xConfiguration SIP Authentication UserName .....	56	xConfiguration UserManagement LDAP Mode .....	64
xConfiguration SIP Authentication Password.....	56	xConfiguration UserManagement LDAP Server Address.....	64
xConfiguration SIP DefaultTransport .....	57	xConfiguration UserManagement LDAP Server Port.....	64
xConfiguration SIP DisplayName .....	57	xConfiguration UserManagement LDAP Encryption .....	64
xConfiguration SIP Ice Mode.....	57	xConfiguration UserManagement LDAP MinimumTLSVersion .....	65
xConfiguration SIP Ice DefaultCandidate.....	57	xConfiguration UserManagement LDAP VerifyServerCertificate .....	65
xConfiguration SIP ListenPort .....	58	xConfiguration UserManagement LDAP Admin Filter .....	65
xConfiguration SIP Proxy [1..4] Address.....	58	xConfiguration UserManagement LDAP Admin Group .....	65
xConfiguration SIP Turn Server.....	58	xConfiguration UserManagement LDAP Attribute .....	66
xConfiguration SIP Turn UserName .....	58	<b>Video configuration</b> .....	<b>66</b>
xConfiguration SIP Turn Password.....	58	xConfiguration Video DefaultMainSource .....	66
xConfiguration SIP URI.....	59	xConfiguration Video Input Connector [1..2] CameraControl Mode .....	66
<b>Standby configuration</b> .....	<b>59</b>	xConfiguration Video Input Connector [1..2] CameraControl Camerald .....	67
xConfiguration Standby Control.....	59	xConfiguration Video Input Connector [1..2] InputSourceType .....	67
xConfiguration Standby Delay.....	59	xConfiguration Video Input Connector [1..2] Name.....	67
<b>SystemUnit configuration</b> .....	<b>60</b>	xConfiguration Video Input Connector [2] Quality.....	67
xConfiguration SystemUnit Name .....	60	xConfiguration Video Input Connector [2] PresentationSelection.....	68
<b>Time configuration</b> .....	<b>60</b>	xConfiguration Video Input Connector [1..2] Visibility .....	68
xConfiguration Time TimeFormat.....	60	xConfiguration Video Monitors.....	68
xConfiguration Time DateFormat .....	60	xConfiguration Video Output Connector [2] CEC Mode.....	69
xConfiguration Time Zone.....	61	xConfiguration Video Output Connector [2] OverscanLevel .....	69
<b>UserInterface configuration</b> .....	<b>62</b>	xConfiguration Video Output Connector [1..2] Resolution .....	69
xConfiguration UserInterface ContactInfo Type.....	62	xConfiguration Video Presentation DefaultSource.....	69
xConfiguration UserInterface KeyTones Mode.....	63	xConfiguration Video Selfview Default Mode .....	69
xConfiguration UserInterface Language .....	63	xConfiguration Video Selfview Default FullscreenMode.....	70
xConfiguration UserInterface OSD Output.....	63	xConfiguration Video Selfview Default OnMonitorRole.....	70
		xConfiguration Video Selfview Default PIPPosition.....	70
		xConfiguration Video Selfview OnCall Mode.....	71
		xConfiguration Video Selfview OnCall Duration.....	71

## Audio configuration

### xConfiguration Audio DefaultVolume

Define the default volume for the speakers. The volume is set to this value when you switch on or restart the video system. Use the controls on the video system to change the volume while it is running. You may also use API commands (xCommand Audio Volume) to change the volume while the video system is running, and to reset to default value.

Requires user role: ADMIN, USER

Default value: 50

#### USAGE:

xConfiguration Audio DefaultVolume: DefaultVolume

where

DefaultVolume: Integer (0..100)

Select a value between 1 and 100. This corresponds to the dB range from -34.5 dB to 15 dB, in steps of 0.5 dB). If set to 0 the audio is switched off.

### xConfiguration Audio Microphones Mute Enabled

Define the microphone mute behaviour on the video system.

Requires user role: ADMIN

Default value: True

#### USAGE:

xConfiguration Audio Microphones Mute Enabled: Enabled

where

Enabled: 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.

### xConfiguration Audio SoundsAndAlerts RingTone

Define which ringtone to use for incoming calls.

Requires user role: ADMIN, USER

Default value: Sunrise

#### USAGE:

xConfiguration Audio SoundsAndAlerts RingTone: RingTone

where

RingTone: Sunrise/Mischief/Ripples/Reflections/Vibes/Delight/Evolve/Playful/Ascent/Calculation/Mellow/Ringer

Select a ringtone from the list.

### xConfiguration Audio SoundsAndAlerts RingVolume

Define the ring volume for incoming calls.

Requires user role: ADMIN, USER

Default value: 50

#### USAGE:

xConfiguration Audio SoundsAndAlerts RingVolume: RingVolume

where

RingVolume: Integer (0..100)

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

### xConfiguration Audio Input MicrophoneMode

This setting only applies to DX80.

The DX80 has microphones in both legs. If you set the microphone mode to Focused, the microphones can be combined to focus sound sensitivity. As a result, the noise in the room is suppressed, and you can be heard better when sitting right in front of the video system. The voice of people not sitting right in front of the system will be suppressed.

If you set the microphone mode to Wide, the system behaves like any other system. The voice of people sitting beside you will be heard, and also more noise from the room.

We recommend that you use Focused mode when you are the only speaker. Use Wide mode when several speakers are in front of the system.

Requires user role: ADMIN

Default value: Wide

#### USAGE:

xConfiguration Audio Input MicrophoneMode: MicrophoneMode

where

MicrophoneMode: Focused/Wide

Focused: Focused sound sensitivity, suppressing sound from sources that are not right in front of the video system.

Wide: Default microphone operation with normal sound sensitivity.

## Conference configuration

### xConfiguration Conference AutoAnswer Mode

Define 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

Default value: Off

#### USAGE:

xConfiguration Conference AutoAnswer Mode: Mode

where

Mode: Off/On

Off: You must answer incoming calls manually by tapping Answer.

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.

### xConfiguration Conference AutoAnswer Mute

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

Requires user role: ADMIN

Default value: Off

#### USAGE:

xConfiguration Conference AutoAnswer Mute: Mute

where

Mute: Off/On

Off: The incoming call will not be muted.

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

### xConfiguration Conference 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

Default value: 0

#### USAGE:

xConfiguration Conference AutoAnswer Delay: Delay

where

Delay: Integer (0..50)

The auto answer delay (seconds).

### xConfiguration Conference DefaultCall Rate

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

Requires user role: ADMIN

Default value: 3072

#### USAGE:

xConfiguration Conference DefaultCall Rate: Rate

where

Rate: Integer (64..3072)

The default call rate (kbps).

### xConfiguration Conference 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. The default value is 60 minutes.

Requires user role: ADMIN

Default value: 60

#### USAGE:

xConfiguration Conference DoNotDisturb DefaultTimeout: DefaultTimeout

where

DefaultTimeout: Integer (1..1440)

The number of minutes (maximum 1440 minutes = 24 hours) before the Do Not Disturb session times out automatically.

### xConfiguration Conference 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

Default value: BestEffort

#### USAGE:

xConfiguration Conference Encryption Mode: Mode

where

Mode: 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.

### xConfiguration Conference 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

Default value: On

#### USAGE:

xConfiguration Conference FarEndControl Mode: Mode

where

Mode: 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.

### xConfiguration Conference MaxReceiveCallRate

Define 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

Default value: 3072

#### USAGE:

xConfiguration Conference MaxReceiveCallRate: MaxReceiveCallRate

where

MaxReceiveCallRate: Integer (64..3072)

The maximum receive call rate (kbps).

### xConfiguration Conference MaxTransmitCallRate

Define 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

Default value: 3072

#### USAGE:

xConfiguration Conference MaxTransmitCallRate: MaxTransmitCallRate

where

MaxTransmitCallRate: Integer (64..3072)

The maximum transmitt call rate (kbps).

### xConfiguration Conference MaxTotalReceiveCallRate

Define the maximum overall receive bit rate allowed. This product does not support multiple simultaneous calls, so the total receive call rate will be the same as the receive bit rate for one call (ref. Conference MaxReceiveCallRate setting).

Requires user role: ADMIN

Default value: 3072

#### USAGE:

xConfiguration Conference MaxTotalReceiveCallRate: MaxTotalReceiveCallRate

where

MaxTotalReceiveCallRate: Integer (64..3072)

The maximum receive call rate (kbps).



### xConfiguration Conference MaxTotalTransmitCallRate

Define the maximum overall transmit bit rate allowed. This product does not support multiple simultaneous calls, so the total transmit call rate will be the same as the transmit bit rate for one call (ref. Conference MaxTransmitCallRate setting).

Requires user role: ADMIN

Default value: 3072

#### USAGE:

xConfiguration Conference MaxTotalTransmitCallRate: MaxTotalTransmitCallRate

where

MaxTotalTransmitCallRate: Integer (64..3072)

The maximum transmit call rate (kbps).

## H323 configuration

### xConfiguration H323 Authentication Mode

Define the authenticatin mode for the H.323 profile.

Requires user role: ADMIN

Default value: Off

#### USAGE:

xConfiguration H323 Authentication Mode: Mode

where

Mode: Off/On

Off: The system will not try to authenticate itself to a H.323 Gatekeeper, but will still try a normal registration.

On: If an H.323 Gatekeeper indicates that it requires authentication, the system will try to authenticate itself to the gatekeeper. Requires the H323 Authentication LoginName and H323 Authentication Password settings to be defined on both the codec and the Gatekeeper.

### xConfiguration H323 Authentication LoginName

The system sends the H323 Authentication Login Name and the H323 Authentication Password to an 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 Authentication Mode to be enabled.

Requires user role: ADMIN

Default value: ""

#### USAGE:

xConfiguration H323 Authentication LoginName: "LoginName"

where

LoginName: String (0, 50)

The authentication login name.

### xConfiguration H323 Authentication Password

The system sends the H323 Authentication Login Name and the H323 Authentication Password to an 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 Authentication Mode to be enabled.

Requires user role: ADMIN

Default value: ""

#### USAGE:

xConfiguration H323 Authentication Password: "Password"

where

Password: String (0, 50)

The authentication password.

### xConfiguration H323 CallSetup Mode

Defines whether to use a Gatekeeper or Direct calling when establishing H.323 calls. Direct H.323 calls can be made also when H323 CallSetup Mode is set to Gatekeeper.

Requires user role: ADMIN

Default value: Gatekeeper

#### USAGE:

xConfiguration H323 CallSetup Mode: Mode

where

Mode: Direct/Gatekeeper

Direct: You can only make an H.323 call by dialing an IP address directly.

Gatekeeper: The system uses a Gatekeeper to make an H.323 call. When choosing this option, the H323 Gatekeeper Address must also be configured.

### xConfiguration H323 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

Default value: Min1024bit

#### USAGE:

xConfiguration H323 Encryption KeySize: KeySize

where

KeySize: Min1024bit/Max1024bit/Min2048bit

Min1024bit: The minimum size is 1024 bit.

Max1024bit: The maximum size is 1024 bit.

Min2048bit: The minimum size is 2048 bit.

### xConfiguration H323 Gatekeeper Address

Define the IP address of the Gatekeeper. Requires H323 CallSetup Mode to be set to Gatekeeper.

Requires user role: ADMIN

Default value: ""

#### USAGE:

xConfiguration H323 Gatekeeper Address: "Address"

where

Address: String (0, 255)

A valid IPv4 address, IPv6 address or DNS name.

### xConfiguration H323 H323Alias E164

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

Requires user role: ADMIN

Default value: ""

#### USAGE:

xConfiguration H323 H323Alias E164: "E164"

where

E164: String (0, 30)

The H.323 Alias E.164 address. Valid characters are 0-9, \* and #.

### xConfiguration H323 H323Alias ID

Define 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.

Requires user role: ADMIN

Default value: ""

#### USAGE:

xConfiguration H323 H323Alias ID: "ID"

where

ID: String (0, 49)

The H.323 Alias ID. Example: "firstname.lastname@company.com", "My H.323 Alias ID"

### 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

Default value: Off

#### USAGE:

xConfiguration H323 NAT Mode: Mode

where

Mode: 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.

### xConfiguration H323 NAT Address

Define 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

Default value: ""

#### USAGE:

xConfiguration H323 NAT Address: "Address"

where

Address: String (0, 64)

A valid IPv4 address or IPv6 address.

## Network configuration

### xConfiguration Network [1] DNS Domain Name

The 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, USER

Default value: ""

#### USAGE:

xConfiguration Network n DNS Domain Name: "Name"

where

n: Index that identifies the network. Range: 1..1

Name: String (0, 64)

The DNS domain name.

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

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

Requires user role: ADMIN, USER

Default value: ""

#### USAGE:

xConfiguration Network n DNS Server m Address: "Address"

where

n: Index that identifies the network. Range: 1..1

m: Index that identifies the DNS server. Maximum three DNS servers are allowed. Range: 1..3

Address: String (0, 64)

A valid IPv4 address or IPv6 address.

### xConfiguration Network [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, USER

Default value: Off

#### USAGE:

xConfiguration Network n IEEE8021X Mode: Mode

where

n: Index that identifies the network. Range: 1..1

Mode: Off/On

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

On: The 802.1X authentication is enabled.

### xConfiguration Network [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] IEEE8021X Eap Tls is enabled (On).

Requires user role: ADMIN, USER

Default value: Off

#### USAGE:

xConfiguration Network n IEEE8021X TlsVerify: TlsVerify

where

n: Index that identifies the network. Range: 1..1

TlsVerify: 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.

### xConfiguration Network [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, USER

Default value: Off

#### USAGE:

xConfiguration Network n IEEE8021X UseClientCertificate: UseClientCertificate

where

n: Index that identifies the network. Range: 1..1

UseClientCertificate: 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.

### xConfiguration Network [1] IEEE8021X Identity

Define the user name for 802.1X authentication.

Requires user role: ADMIN, USER

Default value: ""

#### USAGE:

xConfiguration Network n IEEE8021X Identity: "Identity"

where

n: Index that identifies the network. Range: 1..1

Identity: String (0, 64)

The user name for 802.1X authentication.

### xConfiguration Network [1] IEEE8021X Password

Define the password for 802.1X authentication.

Requires user role: ADMIN, USER

Default value: ""

#### USAGE:

xConfiguration Network *n* IEEE8021X Password: "*Password*"

where

*n*: Index that identifies the network. Range: 1..1

*Password*: String (0, 32)

The password for 802.1X authentication.

### xConfiguration Network [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, USER

Default value: ""

#### USAGE:

xConfiguration Network *n* IEEE8021X AnonymousIdentity: "*AnonymousIdentity*"

where

*n*: Index that identifies the network. Range: 1..1

*AnonymousIdentity*: String (0, 64)

The 802.1X Anonymous ID string.

### xConfiguration Network [1] IEEE8021X Eap Md5

Define 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, USER

Default value: On

#### USAGE:

xConfiguration Network *n* IEEE8021X Eap Md5: *Md5*

where

*n*: Index that identifies the network. Range: 1..1

*Md5*: Off/On

Off: The EAP-MD5 protocol is disabled.

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

### xConfiguration Network [1] IEEE8021X Eap Ttls

Define 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, USER

Default value: On

#### USAGE:

xConfiguration Network *n* IEEE8021X Eap Ttls: *Ttls*

where

*n*: Index that identifies the network. Range: 1..1

*Ttls*: Off/On

Off: The EAP-TTLS protocol is disabled.

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

### xConfiguration Network [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, USER

Default value: On

#### USAGE:

xConfiguration Network n IEEE8021X Eap Tls: Tls

where

n: Index that identifies the network. Range: 1..1

Tls: Off/On

Off: The EAP-TLS protocol is disabled.

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

### xConfiguration Network [1] IEEE8021X Eap Peap

Define 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, USER

Default value: On

#### USAGE:

xConfiguration Network n IEEE8021X Eap Peap: Peap

where

n: Index that identifies the network. Range: 1..1

Peap: Off/On

Off: The EAP-PEAP protocol is disabled.

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

### xConfiguration Network [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, USER

Default value: Dual

#### USAGE:

xConfiguration Network n IPStack: IPStack

where

n: Index that identifies the network. Range: 1..1

IPStack: 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.

### xConfiguration Network [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, USER

Default value: DHCP

#### USAGE:

xConfiguration Network n IPv4 Assignment: Assignment

where

n: Index that identifies the network. Range: 1..1

Assignment: 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.

### xConfiguration Network [1] IPv4 Address

Define the static IPv4 network address for the system. Applicable only when Network IPv4 Assignment is set to Static.

Requires user role: ADMIN, USER

Default value: ""

#### USAGE:

xConfiguration Network *n* IPv4 Address: "*Address*"

where

*n*: Index that identifies the network. Range: 1..1

*Address*: String (0, 64)

A valid IPv4 address.

### xConfiguration Network [1] IPv4 Gateway

Define the IPv4 network gateway address. Applicable only when the Network IPv4 Assignment is set to Static.

Requires user role: ADMIN, USER

Default value: ""

#### USAGE:

xConfiguration Network *n* IPv4 Gateway: "*Gateway*"

where

*n*: Index that identifies the network. Range: 1..1

*Gateway*: String (0, 64)

A valid IPv4 address.

### xConfiguration Network [1] IPv4 SubnetMask

Define the IPv4 network subnet mask. Applicable only when the Network IPv4 Assignment is set to Static.

Requires user role: ADMIN, USER

Default value: ""

#### USAGE:

xConfiguration Network *n* IPv4 SubnetMask: "*SubnetMask*"

where

*n*: Index that identifies the network. Range: 1..1

*SubnetMask*: String (0, 64)

A valid IPv4 address.

### xConfiguration Network [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, USER

Default value: Autoconf

#### USAGE:

xConfiguration Network *n* IPv6 Assignment: "*Assignment*"

where

*n*: Index that identifies the network. Range: 1..1

*Assignment*: 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.



### xConfiguration Network [1] IPv6 Address

Define the static IPv6 network address for the system. Applicable only when the Network IPv6 Assignment is set to Static.

Requires user role: ADMIN, USER

Default value: ""

#### USAGE:

xConfiguration Network *n* IPv6 Address: "*Address*"

where

*n*: Index that identifies the network. Range: 1..1

*Address*: String (0, 64)

A valid IPv6 address including a network mask. Example: 2001:DB8::/48

### xConfiguration Network [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, USER

Default value: ""

#### USAGE:

xConfiguration Network *n* IPv6 Gateway: "*Gateway*"

where

*n*: Index that identifies the network. Range: 1..1

*Gateway*: String (0, 64)

A valid IPv6 address.

### xConfiguration Network [1] IPv6 DHCPOptions

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

Requires user role: ADMIN, USER

Default value: On

#### USAGE:

xConfiguration Network *n* IPv6 DHCPOptions: "*DHCPOptions*"

where

*n*: Index that identifies the network. Range: 1..1

*DHCPOptions*: 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.

### xConfiguration Network [1] MTU

Define the Ethernet MTU (Maximum Transmission Unit) size. The MTU size must be supported by your network infrastructure. The minimum size is 576 for IPv4 and 1280 for IPv6.

Requires user role: ADMIN, USER

Default value: 1500

#### USAGE:

xConfiguration Network *n* MTU: *MTU*

where

*n*: Index that identifies the network. Range: 1..1

*MTU*: Integer (576..1500)

Set a value for the MTU (bytes).

### xConfiguration Network [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, USER

Default value: Diffserv

#### USAGE:

xConfiguration Network *n* QoS Mode: *Mode*

where

*n*: Index that identifies the network. Range: 1..1

*Mode*: 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.

### xConfiguration Network [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, USER

Default value: 0

#### USAGE:

xConfiguration Network *n* QoS DiffServ Audio: *Audio*

where

*n*: Index that identifies the network. Range: 1..1

*Audio*: Integer (0..63)

Set the priority of the audio packets in the IP network - the higher the number, the higher the priority. The default value is 0 (best effort).

### xConfiguration Network [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, USER

Default value: 0

#### USAGE:

xConfiguration Network *n* QoS Diffserv Video: *Video*

where

*n*: Index that identifies the network. Range: 1..1

*Video*: Integer (0..63)

Set the priority of the video packets in the IP network - the higher the number, the higher the priority. The default value is 0 (best effort).

### xConfiguration Network [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, USER

Default value: 0

#### USAGE:

xConfiguration Network *n* QoS Diffserv Data: *Data*

where

*n*: Index that identifies the network. Range: 1..1

*Data*: Integer (0..63)

Set the priority of the data packets in the IP network - the higher the number, the higher the priority. The default value is 0 (best effort).

### xConfiguration Network [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, USER

Default value: 0

#### USAGE:

xConfiguration Network n QoS Diffserv Signalling: Signalling

where

n: Index that identifies the network. Range: 1..1

Signalling: Integer (0..63)

Set the priority of the signalling packets in the IP network - the higher the number, the higher the priority. The default value is 0 (best effort).

### xConfiguration Network [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, USER

Default value: 0

#### USAGE:

xConfiguration Network n QoS Diffserv ICMPv6: ICMPv6

where

n: Index that identifies the network. Range: 1..1

ICMPv6: Integer (0..63)

Set the priority of the ICMPv6 packets in the IP network - the higher the number, the higher the priority. The default value is 0 (best effort).

### xConfiguration Network [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, USER

Default value: 0

#### USAGE:

xConfiguration Network n QoS Diffserv NTP: NTP

where

n: Index that identifies the network. Range: 1..1

NTP: Integer (0..63)

Set the priority of the NTP packets in the IP network - the higher the number, the higher the priority. The default value is 0 (best effort).

### xConfiguration Network [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, USER

Default value: ""

#### USAGE:

xConfiguration Network n RemoteAccess Allow: "Allow"

where

n: Index that identifies the network. Range: 1..1

Allow: String (0, 255)

A valid IPv4 address or IPv6 address.

### xConfiguration Network [1] Speed

Define the Ethernet link speed. We recommend not to change from the default value, which negotiates with the network to set the speed automatically. If you do not use autonegotiation, make sure that the speed you choose is supported by the closest switch in your network infrastructure.

Requires user role: ADMIN, USER

Default value: Auto

#### USAGE:

xConfiguration Network n Speed: Speed

where

n: Index that identifies the network. Range: 1..1

Speed: 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.

### xConfiguration Network [1] VLAN Voice Mode

Define 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, USER

Default value: Auto

#### USAGE:

`xConfiguration Network n VLAN Voice Mode: Mode`

where

n: Index that identifies the network. Range: 1..1

Mode: 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.

### xConfiguration Network [1] VLAN Voice VlanId

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

Requires user role: ADMIN, USER

Default value: 1

#### USAGE:

`xConfiguration Network n VLAN Voice VlanId: VlanId`

where

n: Index that identifies the network. Range: 1..1

VlanId: Integer (1..4094)

Set the VLAN voice ID.

## 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

Default value: On

#### USAGE:

`xConfiguration NetworkServices CDP Mode: Mode`

where

Mode: Off/On

Off: The CDP daemon is disabled.

On: The CDP daemon is enabled.

### xConfiguration NetworkServices H323 Mode

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

Requires user role: ADMIN

Default value: Off

#### USAGE:

`xConfiguration NetworkServices H323 Mode: Mode`

where

Mode: 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).

### xConfiguration NetworkServices HTTP Mode

Define whether or not to allow access to the video system using the HTTP or HTTPS (HTTP Secure) protocols. Note that the video system's web interface use HTTP or HTTPS. If this setting is switched Off, you cannot use the web interface.

If you need extra security (encryption and decryption of requests, and pages that are returned by the web server), allow only HTTPS.

Requires user role: ADMIN

Default value: HTTP+HTTPS

#### USAGE:

xConfiguration NetworkServices HTTP Mode: Mode

where

Mode: Off/HTTP+HTTPS/HTTPS

Off: Access to the video system not allowed via HTTP or HTTPS.

HTTP+HTTPS: Access to the video system allowed via both HTTP and HTTPS.

HTTPS: Access to the video system allowed via HTTPS, but not via HTTP.

### 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

Default value: Off

#### USAGE:

xConfiguration NetworkServices HTTPS VerifyServerCertificate:  
VerifyServerCertificate

where

VerifyServerCertificate: 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.

### 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

Default value: Off

#### USAGE:

xConfiguration NetworkServices HTTPS VerifyClientCertificate:  
VerifyClientCertificate

where

VerifyClientCertificate: 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.

### 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

Default value: Auto

#### USAGE:

xConfiguration NetworkServices NTP Mode: Mode

where

Mode: 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 Server [n] Address setting will be used.

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

Off: The system will not use an NTP server. The NetworkServices NTP Server [n] Address setting will be ignored.

### xConfiguration NetworkServices NTP Server [1..3] 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

Default value: 0.tandberg.pool.ntp.org

#### USAGE:

xConfiguration NetworkServices NTP Server *n* Address: "*Address*"

where

*n*: Unique ID that identifies the NTP server. Range: 1..3

*Address*: String (0, 255)

A valid IPv4 address, IPv6 address or DNS name.

### xConfiguration NetworkServices SIP Mode

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

Requires user role: ADMIN

Default value: On

#### USAGE:

xConfiguration NetworkServices SIP Mode: *Mode*

where

*Mode*: Off/On

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

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

### 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

Default value: ReadOnly

#### USAGE:

xConfiguration NetworkServices SNMP Mode: *Mode*

where

*Mode*: 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.

### xConfiguration NetworkServices SNMP Host [1..3] Address

Define 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

Default value: ""

#### USAGE:

xConfiguration NetworkServices SNMP Host *n* Address: "*Address*"

where

*n*: Identifies the SNMP host. Maximum three SNMP hosts are allowed. Range: 1..3

*Address*: String (0, 255)

A valid IPv4 address, IPv6 address or DNS name.



### xConfiguration NetworkServices SNMP CommunityName

Define 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

Default value: ""

#### USAGE:

xConfiguration NetworkServices SNMP CommunityName: "CommunityName"

where

CommunityName: String (0, 50)

The SNMP community name.

### xConfiguration NetworkServices SNMP SystemContact

Define the name of the Network Services SNMP System Contact.

Requires user role: ADMIN

Default value: ""

#### USAGE:

xConfiguration NetworkServices SNMP SystemContact: "SystemContact"

where

SystemContact: String (0, 50)

The name of the SNMP system contact.

### xConfiguration NetworkServices SNMP SystemLocation

Define the name of the Network Services SNMP System Location.

Requires user role: ADMIN

Default value: ""

#### USAGE:

xConfiguration NetworkServices SNMP SystemLocation: "SystemLocation"

where

SystemLocation: String (0, 50)

The name of the SNMP system location.

### 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

Default value: On

#### USAGE:

xConfiguration NetworkServices SSH Mode: Mode

where

Mode: Off/On

Off: The SSH protocol is disabled.

On: The SSH protocol is enabled.

### xConfiguration NetworkServices Telnet Mode

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

Requires user role: ADMIN

Default value: Off

#### USAGE:

xConfiguration NetworkServices Telnet Mode: Mode

where

Mode: Off/On

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

On: The Telnet protocol is enabled.

### xConfiguration NetworkServices WelcomeText

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

Requires user role: ADMIN

Default value: On

#### USAGE:

xConfiguration NetworkServices WelcomeText: WelcomeText

where

WelcomeText: 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.

## Peripherals configuration

### xConfiguration Peripherals Profile ControlSystems

Define if a third-party control system, for example Crestron or AMX, is expected to be connected to the video system. This information is used by the video system's diagnostics service. If the number of connected control systems does not match this setting, the diagnostics service will report it as an inconsistency. Note that only one third-party control system is supported.

If set to 1, the control system must send heart beats to the video system using xCommand Peripherals Pair and HeartBeat commands. Failing to do so will cause the in-room control extensions to show a warning that the video system has lost connectivity to the control system.

Requires user role: ADMIN

Default value: NotSet

#### USAGE:

xConfiguration Peripherals Profile ControlSystems: ControlSystems

where

ControlSystems: 1/NotSet

1: One third-party control system should be connected to the video system.

NotSet: No check for a third-party control system is performed.

## Phonebook configuration

### xConfiguration Phonebook Server [1] ID

Define a name for the external phone book.

Requires user role: ADMIN

Default value: ""

#### USAGE:

xConfiguration Phonebook Server n ID: "ID"

where

n: Unique ID that identifies the phonebook server. Range: 1..1

ID: String (0, 64)

The name for the external phone book.

### xConfiguration Phonebook Server [1] Type

Select the phonebook server type.

Requires user role: ADMIN

Default value: Off

#### USAGE:

xConfiguration Phonebook Server n Type: Type

where

n: Unique ID that identifies the phonebook server. Range: 1..1

Type: Off/CUCM/Spark/TMS/VCS

Off: Do not use a phonebook.

CUCM: The phonebook is located on the Cisco Unified Communications Manager.

Spark: The phonebook is located on Spark.

TMS: The phonebook is located on the Cisco TelePresence Management Suite server.

VCS: The phonebook is located on the Cisco TelePresence Video Communication Server.

### xConfiguration Phonebook Server [1] URL

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

Requires user role: ADMIN

Default value: ""

#### USAGE:

xConfiguration Phonebook Server n URL: "URL"

where

n: Unique ID that identifies the phonebook server. Range: 1..1

URL: String (0, 255)

A valid address (URL) to the phone book server.

## 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, USER

Default value: Auto

#### USAGE:

xConfiguration Provisioning Connectivity: Connectivity

where

Connectivity: 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.

### 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, USER

Default value: Auto

#### USAGE:

xConfiguration Provisioning Mode: Mode

where

Mode: Off/Auto/CUCM/Edge/Spark/TMS/VCS

Off: The video system is not configured by a provisioning system.

Auto: Automatically select the provisioning server.

CUCM: Push configurations to the video system from CUCM (Cisco Unified Communications Manager).

Edge: Push configurations to the video system from CUCM (Cisco Unified Communications Manager). The system connects to CUCM via the Collaboration Edge infrastructure.

Spark: Push configurations to the video system from Spark.

TMS: Push configurations to the video system from TMS (Cisco TelePresence Management System).

VCS: Push configurations to the video system from VCS (Cisco TelePresence Video Communication Server).

### xConfiguration Provisioning LoginName

This is the username part of the credentials used to authenticate the video system with the provisioning server. This setting must be used when required by the provisioning server.

Requires user role: ADMIN, USER

Default value: ""

#### USAGE:

xConfiguration Provisioning LoginName: "LoginName"

where

LoginName: String (0, 80)

A valid username.

### xConfiguration Provisioning Password

This is the password part of the credentials used to authenticate the video system with the provisioning server. This setting must be used when required by the provisioning server.

Requires user role: ADMIN, USER

Default value: ""

#### USAGE:

xConfiguration Provisioning Password: "Password"

where

Password: String (0, 64)

A valid password.

### xConfiguration Provisioning HttpMethod

Select the HTTP method to be used for the provisioning.

Requires user role: ADMIN, USER

Default value: POST

#### USAGE:

xConfiguration Provisioning HttpMethod: HttpMethod

where

HttpMethod: GET/POST

GET: Select GET when the provisioning server supports GET.

POST: Select POST when the provisioning server supports POST.

### xConfiguration Provisioning ExternalManager Address

Define 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, USER

Default value: ""

#### USAGE:

xConfiguration Provisioning ExternalManager Address: "Address"

where

Address: String (0, 64)

A valid IPv4 address, IPv6 address or DNS name.

### 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. Define 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, USER

Default value: ""

#### USAGE:

xConfiguration Provisioning ExternalManager AlternateAddress: "AlternateAddress"

where

AlternateAddress: String (0, 64)

A valid IPv4 address, IPv6 address or DNS name.

### xConfiguration Provisioning ExternalManager Protocol

Define whether to use the HTTP (unsecure communication) or HTTPS (secure communication) protocol when sending requests to the external manager / provisioning system.

The chosen protocol must be enabled in the NetworkServices HTTP Mode setting.

Requires user role: ADMIN, USER

Default value: HTTP

#### USAGE:

xConfiguration Provisioning ExternalManager Protocol: "Protocol"

where

Protocol: HTTPS/HTTP

HTTPS: Send requests via HTTPS.

HTTP: Send requests via HTTP.

### xConfiguration Provisioning ExternalManager Path

Define 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, USER

Default value: ""

#### USAGE:

xConfiguration Provisioning ExternalManager Path: "Path"

where

Path: String (0, 255)

A valid path to the external manager or provisioning system.

### xConfiguration Provisioning ExternalManager Domain

Define the SIP domain for the VCS provisioning server.

Requires user role: ADMIN, USER

Default value: ""

#### USAGE:

xConfiguration Provisioning ExternalManager Domain: "Domain"

where

Domain: String (0, 64)

A valid domain name.

## Proximity configuration

### xConfiguration Proximity Mode

Determine whether the video system will emit ultrasound pairing messages or not.

When the video system emits ultrasound, Proximity clients can detect that they are close to the video system. In order to use a client, at least one of the Proximity services must be enabled (refer to the Proximity Services settings). In general, Cisco recommends enabling all the Proximity services.

Requires user role: ADMIN, USER

Default value: Off

#### USAGE:

xConfiguration Proximity Mode: Mode

where

Mode: Off/On

Off: The video system does not emit ultrasound, and Proximity services cannot be used.

On: The video system emits ultrasound, and Proximity clients can detect that they are close to the video system. Enabled Proximity services can be used.

### xConfiguration Proximity Services CallControl

Enable or disable basic call control features on Proximity clients. When this setting is enabled, you are able to control a call using a Proximity client (for example dial, mute, adjust volume and hang up). This service is supported by mobile devices (iOS and Android). Proximity Mode must be On for this setting to take any effect.

Requires user role: ADMIN, USER

Default value: Disabled

#### USAGE:

xConfiguration Proximity Services CallControl: CallControl

where

CallControl: Enabled/Disabled

Enabled: Call control from a Proximity client is enabled.

Disabled: Call control from a Proximity client is disabled.

### xConfiguration Proximity Services ContentShare FromClients

Enable or disable content sharing from Proximity clients. When this setting is enabled, you can share content from a Proximity client wirelessly on the video system, e.g. share your laptop screen. This service is supported by laptops (OS X and Windows). Proximity Mode must be On for this setting to take any effect.

Requires user role: ADMIN, USER

Default value: Enabled

#### USAGE:

xConfiguration Proximity Services ContentShare FromClients: FromClients

where

FromClients: Enabled/Disabled

Enabled: Content sharing from a Proximity client is enabled.

Disabled: Content sharing from a Proximity client is disabled.

### xConfiguration Proximity Services ContentShare ToClients

Enable or disable content sharing to Proximity clients. When enabled, Proximity clients will receive the presentation from the video system. You can zoom in on details, view previous content and take snapshots. This service is supported by mobile devices (iOS and Android). Proximity Mode must be On for this setting to take any effect.

Requires user role: ADMIN, USER

Default value: Disabled

#### USAGE:

xConfiguration Proximity Services ContentShare ToClients: ToClients

where

ToClients: Enabled/Disabled

Enabled: Content sharing to a Proximity client is enabled.

Disabled: Content sharing to a Proximity client is disabled.

## SerialPort configuration

### xConfiguration SerialPort Mode

Enable/disable the serial port (connection via Micro USB to USB cable). The serial port uses 115200 bps, 8 data bits, no parity and 1 stop bit.

Requires user role: ADMIN

Default value: On

#### USAGE:

xConfiguration SerialPort Mode: Mode

where

Mode: Off/On

Off: Disable the serial port.

On: Enable the serial port.

### xConfiguration SerialPort LoginRequired

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

Requires user role: ADMIN

Default value: On

#### USAGE:

xConfiguration SerialPort LoginRequired: LoginRequired

where

LoginRequired: 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.

## SIP configuration

### xConfiguration SIP Authentication UserName

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

Requires user role: ADMIN

Default value: ""

#### USAGE:

xConfiguration SIP Authentication UserName: "UserName"

where

UserName: String (0, 128)

A valid username.

### xConfiguration SIP Authentication Password

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

Requires user role: ADMIN

Default value: ""

#### USAGE:

xConfiguration SIP Authentication Password: "Password"

where

Password: String (0, 128)

A valid password.



### xConfiguration SIP DefaultTransport

Select the transport protocol to be used over the LAN.

Requires user role: ADMIN

Default value: Auto

#### USAGE:

xConfiguration SIP DefaultTransport: DefaultTransport

where

DefaultTransport: 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.

### xConfiguration SIP DisplayName

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

Requires user role: ADMIN

Default value: ""

#### USAGE:

xConfiguration SIP DisplayName: "DisplayName"

where

DisplayName: String (0, 550)

The name to be displayed instead of the SIP URI.

### xConfiguration SIP Ice Mode

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

Requires user role: ADMIN

Default value: Auto

#### USAGE:

xConfiguration SIP Ice Mode: Mode

where

Mode: Auto/Off/On

Auto: ICE is enabled if a TURN server is provided, otherwise ICE is disabled.

Off: ICE is disabled.

On: ICE is enabled.

### xConfiguration SIP Ice DefaultCandidate

The ICE protocol needs some time to reach a conclusion about which media route to use (up to the first 5 seconds of a call). During this period media for the video system will be sent to the Default Candidate as defined in this setting.

Requires user role: ADMIN

Default value: Host

#### USAGE:

xConfiguration SIP Ice DefaultCandidate: DefaultCandidate

where

DefaultCandidate: Host/Rflx/Relay

Host: Send media to the video system's private IP address.

Rflx: Send media to the video system's public IP address, as seen by the TURN server.

Relay: Send media to the IP address and port allocated on the TURN server.

### 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 (CUCCM or VCS). It is recommended to leave this setting at its default value.

Requires user role: ADMIN

Default value: On

#### USAGE:

xConfiguration SIP ListenPort: ListenPort

where

ListenPort: 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.

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

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

Requires user role: ADMIN

Default value: ""

#### USAGE:

xConfiguration SIP Proxy n Address: "Address"

where

n: Index that identifies the proxy (maximum 4 proxys can be defined). Range: 1..4

Address: String (0, 255)

A valid IPv4 address, IPv6 address or DNS name.

### xConfiguration SIP Turn Server

Define the address of the TURN (Traversal Using Relay NAT) server. 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

Default value: ""

#### USAGE:

xConfiguration SIP Turn Server: "Server"

where

Server: String (0, 255)

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

### xConfiguration SIP Turn UserName

Define the user name needed for accessing the TURN server.

Requires user role: ADMIN

Default value: ""

#### USAGE:

xConfiguration SIP Turn UserName: "UserName"

where

UserName: String (0, 128)

A valid user name.

### xConfiguration SIP Turn Password

Define the password needed for accessing the TURN server.

Requires user role: ADMIN

Default value: ""

#### USAGE:

xConfiguration SIP Turn Password: "Password"

where

Password: String (0, 128)

A valid password.

### xConfiguration SIP 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

Default value: ""

#### USAGE:

xConfiguration SIP URI: "URI"

where

URI: String (0, 255)

An address (URI) that is compliant with the SIP URI syntax.

### Standby configuration

#### xConfiguration Standby Control

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

Requires user role: ADMIN

Default value: On

#### USAGE:

xConfiguration Standby Control: Control

where

Control: Off/On

Off: The system will not enter standby mode.

On: The system will enter standby mode when the Standby Delay has timed out.

Requires the Standby Delay to be set to an appropriate value.

#### 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

Default value: 10

#### USAGE:

xConfiguration Standby Delay: Delay

where

Delay: Integer (1..480)

Set the standby delay (minutes).

## 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.

Requires user role: ADMIN

Default value: ""

#### USAGE:

xConfiguration SystemUnit Name: "Name"

where

Name: String (0, 50)

Define the system name.

## Time configuration

### xConfiguration Time TimeFormat

Define the time format.

Requires user role: ADMIN, USER

Default value: 24H

#### USAGE:

xConfiguration Time TimeFormat: TimeFormat

where

TimeFormat: 24H/12H

24H: Set the time format to 24 hours.

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

### xConfiguration Time DateFormat

Define the date format.

Requires user role: ADMIN, USER

Default value: DD\_MM\_YY

#### USAGE:

xConfiguration Time DateFormat: DateFormat

where

DateFormat: 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

## xConfiguration Time Zone

Define 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: ADMIN, USER

Default value: Etc/UTC

### USAGE:

xConfiguration Time Zone: Zone

where

Zone: 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/EI\_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\_Nelson, 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/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/Troll, Antarctica/Vostok, Arctic/Longyearbyen, Asia/Aden, Asia/Almaty, Asia/Amman, Asia/Anadyr, Asia/Aqtou, Asia/Aqtobe, Asia/Ashgabat, Asia/Ashkhabad, Asia/Baghdad, Asia/Bahrain, Asia/Baku, Asia/Bangkok, Asia/Barnaul, Asia/Beirut, Asia/Bishkek, Asia/Brunei, Asia/Calcutta, Asia/Chita, 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/Srednekolymysk, Asia/Taipei, Asia/Tashkent, Asia/Tbilisi, Asia/Tehran, Asia/Tel\_Aviv, Asia/Thimbu, Asia/Thimphu, Asia/Tokyo, Asia/Tomsk, 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/Astrakhan, 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/Kirov, 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/Ulyanovsk, 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/Bougainville, Pacific/Chatham, Pacific/Chuuk, Pacific/Easter, Pacific/Efate, Pacific/Enderbury, Pacific/Fakaofu, 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

Select a time zone from the list.

## UserInterface configuration

### xConfiguration UserInterface ContactInfo Type

Choose which type of contact information to show on the Home screen, and when you tap the Settings icon.

Requires user role: ADMIN

Default value: Auto

#### USAGE:

xConfiguration UserInterface ContactInfo Type: Type

where

Type: Auto/None/IPv4/IPv6/SipUri/SystemName/DisplayName

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

None: Do not show any contact information.

IPv4: Show the system's IPv4 address.

IPv6: Show the system's IPv6 address.

SipUri: Show the system's SIP URI (refer to the SIP URI setting).

SystemName: Show the system's name (refer to the SystemUnit Name setting).

DisplayName: Show the system's display name (refer to the SIP DisplayName setting).

### xConfiguration UserInterface KeyTones Mode

You can configure the system to make a keyboard click sound effect (key tone) when typing text or numbers.

Requires user role: ADMIN, USER

Default value: Off

#### USAGE:

xConfiguration UserInterface KeyTones Mode: Mode

where

Mode: Off/On

Off: There is no key tone sound effect.

On: The key tone sound effect is turned on.

### xConfiguration UserInterface Language

Select the language to be used in menus and messages on the screen.

Requires user role: ADMIN, USER

Default value: English

#### USAGE:

xConfiguration UserInterface Language: Language

where

Language: Arabic/Catalan/ChineseSimplified/ChineseTraditional/Czech/Danish/Dutch/English/EnglishUK/Finnish/French/FrenchCanadian/German/Hebrew/Hungarian/Italian/Japanese/Korean/Norwegian/Polish/PortugueseBrazilian/Russian/Spanish/SpanishLatin/Swedish/Turkish

Select a language from the list.

### xConfiguration UserInterface OSD Output

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

Requires user role: ADMIN

Default value: Auto

#### USAGE:

xConfiguration UserInterface OSD Output: Output

where

Output: Auto

Auto: The system sends the on-screen information and indicators to the system's integrated display.

## UserManagement configuration

### xConfiguration UserManagement LDAP Mode

The video system supports the use of an LDAP (Lightweight Directory Access Protocol) server as a central place to store and validate user names and passwords. Use this setting to configure whether or not to use LDAP authentication. Our implementation is tested for the Microsoft Active Directory (AD) service.

Requires user role: ADMIN

Default value: Off

#### USAGE:

`xConfiguration UserManagement LDAP Mode: Mode`

where

Mode: Off/On

Off: LDAP authentication is not allowed.

On: Use LDAP authentication.

### xConfiguration UserManagement LDAP Server Address

Set the IP address or hostname of the LDAP server.

Requires user role: ADMIN

Default value: ""

#### USAGE:

`xConfiguration UserManagement LDAP Server Address: "Address"`

where

Address: String (0, 255)

A valid IPv4 address, IPv6 address or hostname.

### xConfiguration UserManagement LDAP Server Port

Set the port to connect to the LDAP server on. If set to 0, use the default for the selected protocol (see the UserManagement LDAP Encryption setting).

Requires user role: ADMIN

Default value: 0

#### USAGE:

`xConfiguration UserManagement LDAP Server Port: Port`

where

Port: Integer (0..65535)

The LDAP server port number.

### xConfiguration UserManagement LDAP Encryption

Define how to secure the communication between the video system and the LDAP server. You can override the port number by using the UserManagement LDAP Server Port setting.

Requires user role: ADMIN

Default value: LDAPS

#### USAGE:

`xConfiguration UserManagement LDAP Encryption: Encryption`

where

Encryption: LDAPS/None/STARTTLS

LDAPS: Connect to the LDAP server on port 636 over TLS (Transport Layer Security).

None: Connect to LDAP server on port 389 with no encryption.

STARTTLS: Connect to LDAP server on port 389, then send STARTTLS to enable TLS encryption.



### xConfiguration UserManagement LDAP MinimumTLSVersion

Set the lowest version of the TLS (Transport Layer Security) protocol that is allowed.

Requires user role: ADMIN

Default value: TLSv1.2

#### USAGE:

xConfiguration UserManagement LDAP MinimumTLSVersion: MinimumTLSVersion

where

MinimumTLSVersion: TLSv1.0/TLSv1.1/TLSv1.2

TLSv1.0: Support TLS version 1.0 or higher.

TLSv1.1: Support TLS version 1.1 or higher.

TLSv1.2: Support TLS version 1.2 or higher.

### xConfiguration UserManagement LDAP VerifyServerCertificate

When the video system connects to an LDAP server, the server will identify itself to the video system by presenting its certificate. Use this setting to determine whether or not the video system will verify the server certificate.

Requires user role: ADMIN

Default value: On

#### USAGE:

xConfiguration UserManagement LDAP VerifyServerCertificate:

VerifyServerCertificate

where

VerifyServerCertificate: Off/On

Off: The video system will not verify the LDAP server's certificate.

On: The video system must verify that the LDAP server's certificate is signed by a trusted Certificate Authority (CA). The CA must be on the list of trusted CAs that are uploaded to the system in advance. Use the video system's web interface to manage the list of trusted CAs (see more details in the administrator guide).

### xConfiguration UserManagement LDAP Admin Filter

The LDAP filter is used to determine which users should be granted administrator privileges. If set, this setting takes precedence over the UserManagement LDAP Admin Group setting.

Requires user role: ADMIN

Default value: ""

#### USAGE:

xConfiguration UserManagement LDAP Admin Filter: "Filter"

where

Filter: String (0, 1024)

Refer to the LDAP specification for the syntax of this string. Example: "(CN=adminuser)"

### xConfiguration UserManagement LDAP Admin Group

Members of this AD (Active Directory) group will be given administrator access. This setting is a shorthand for saying (memberOf:1.2.840.113556.1.4.1941:=<group name>). If UserManagement LDAP Admin Filter is set, this setting is ignored.

Requires user role: ADMIN

Default value: ""

#### USAGE:

xConfiguration UserManagement LDAP Admin Group: "Group"

where

Group: String (0, 255)

The distinguishing name of the AD group. Example: "CN=admin\_group, OU=company groups, DC=company, DC=com"

### xConfiguration UserManagement LDAP Attribute

The attribute used to map to the provided username. If not set, sAMAccountName is used.

Requires user role: ADMIN

Default value: ""

#### USAGE:

xConfiguration UserManagement LDAP Attribute: "Attribute"

where

Attribute: String (0, 255)

The attribute name.

## Video configuration

### xConfiguration Video DefaultMainSource

Define which video input source shall be used as the main video source.

Requires user role: ADMIN, USER

Default value: 1

#### USAGE:

xConfiguration Video DefaultMainSource: DefaultMainSource

where

DefaultMainSource: 1

Set the source to be used as the main video source.

### xConfiguration Video Input Connector [1..2] CameraControl Mode

Define whether a camera can be controlled or not. This value is fixed for both Connector 1 (integrated camera) and Connector 2 (HDMI), and cannot be changed.

Requires user role: ADMIN

Default value: Off

#### USAGE:

xConfiguration Video Input Connector n CameraControl Mode: Mode

where

n: Unique ID that identifies the video input connector. Range: 1..2

Mode: Off

Off: Disable camera control.

### xConfiguration Video Input Connector [1..2] CameraControl CameraId

The camera ID is a unique identifier of the cameras that are connected to the video input.

Requires user role: ADMIN

Default value: 1

#### USAGE:

xConfiguration Video Input Connector n CameraControl CameraId: CameraId

where

n: Unique ID that identifies the video input connector. Range: 1..2

CameraId: 1

The camera ID is fixed and cannot be changed.

### xConfiguration Video Input Connector [1..2] InputSourceType

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

Note that Connector 1 is the system's integrated camera.

Requires user role: ADMIN

Default value: Connector 1: camera Connector 2: desktop

#### USAGE:

xConfiguration Video Input Connector n InputSourceType: InputSourceType

where

n: Unique ID that identifies the video input connector. Range: 1..2

InputSourceType: Connector 1: camera Connector 2: camera/desktop/document\_camera/mediaplayer/PC/whiteboard/other

camera: Use this when a camera is connected to the video input.

desktop: Use this when the video system's monitor is the main screen for the PC / laptop that is connected to this input. The Video Input Connector [n] PresentationSelection setting must then also be set to desktop.

document\_camera: Use this when a document camera is connected to the video input.

mediaplayer: Use this when a media player is connected to the video input.

PC: Use this when a computer is connected to the video input.

whiteboard: Use this when a whiteboard camera is connected to the video input.

other: Use this when the other options do not match.

### xConfiguration Video Input Connector [1..2] Name

Define a name for the video input connector.

Requires user role: ADMIN

Default value: ""

#### USAGE:

xConfiguration Video Input Connector n Name: "Name"

where

n: Unique ID that identifies the video input connector. Range: 1..2

Name: String (0, 50)

Name for the video input connector.

### xConfiguration Video Input Connector [2] Quality

When encoding and transmitting video there is 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. This setting specifies whether to give priority to high frame rate or to high resolution.

Requires user role: ADMIN

Default value: Sharpness

#### USAGE:

xConfiguration Video Input Connector n Quality: Quality

where

n: Unique ID that identifies the video input connector. Range: 2..2

Quality: 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.

### xConfiguration Video Input Connector [2] 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. Sharing the presentation with the far end requires additional action (select Share on the user interface) except when this setting is set to AutoShare.

Requires user role: ADMIN

Default value: Desktop

#### USAGE:

xConfiguration Video Input Connector *n* PresentationSelection: PresentationSelection

where

*n*: Unique ID that identifies the video input connector. Range: 2..2

PresentationSelection: AutoShare/Desktop/Manual/OnConnect

**AutoShare:** While in a call, the content on the video input will automatically be presented to the far end as well as on the local screen when you connect the cable, or when the source is activated otherwise (for example when a connected computer wakes up from sleep mode). You do not have to select Share on the user interface. If a presentation source is already connected when you make or answer a call, you have to manually select Share on the user interface.

**Desktop:** The content on the video input will be presented on the screen when you connect the cable, or when the source is activated otherwise (for example when a connected computer wakes up from sleep mode). This applies both when idle and in a call. Also, the content on the video input will stay on the screen when you leave the call, provided that it was the active input at the time of leaving.

**Manual:** The content on the video input will not be presented on the screen until you select Share from the user interface.

**OnConnect:** The content on the video input will be presented on screen when you connect the cable, or when the source is activated otherwise (for example when a connected computer wakes up from sleep mode). Otherwise, the behavior is the same as in manual mode.

### xConfiguration Video Input Connector [1..2] Visibility

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

Note that Connector 1 is the system's integrated camera, which is not available as a presentation source.

The default value is IfSignal for Video Input Connector 2 Visibility (the HDMI connector).

Requires user role: ADMIN

Default value: Connector 1: Never Connector 2: Always Connector 3: OnConnect

#### USAGE:

xConfiguration Video Input Connector *n* Visibility: Visibility

where

*n*: Unique ID that identifies the video input connector. Range: 1..2

Visibility: Connector 1: Never Connector 2, 3: 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.

### xConfiguration Video Monitors

Define the monitor layout mode. Note that this video system supports only one monitor, so this value is fixed and cannot be changed.

Requires user role: ADMIN

Default value: Single

#### USAGE:

xConfiguration Video Monitors: Monitors

where

Monitors: Single

**Single:** The layout is shown on the video system's monitor.

### xConfiguration Video Output Connector [2] CEC Mode

The HDMI output (Output Connector 2) is for future use.

**USAGE:**

xConfiguration Video Output Connector *n* CEC Mode: *Mode*

where

*n*: Unique ID that identifies the video output connector. Range: 2..2

### xConfiguration Video Output Connector [2] OverscanLevel

The HDMI output (Output Connector 2) is for future use.

**USAGE:**

xConfiguration Video Output Connector *n* OverscanLevel: *OverscanLevel*

where

*n*: Unique ID that identifies the video output connector. Range: 2..2

### xConfiguration Video Output Connector [1..2] Resolution

Connector 1: The resolution and refresh rate for the integrated display. This value is fixed and cannot be changed.

Connector 2: The HDMI output (Output Connector 2) is for future use.

Default value: Connector 1: 1920\_1080\_60

**USAGE:**

xConfiguration Video Output Connector *n* Resolution: *Resolution*

where

*n*: Unique ID that identifies the video output connector. Range: 1..2

*Resolution*: Connector 1: 1920\_1080\_60

1920\_1080\_60: The resolution is 1920 x 1080, and the refresh rate is 60 Hz.

### xConfiguration Video Presentation DefaultSource

Define which video input source to use as a default presentation source. This setting may be used by the API and 3rd party user interfaces. It is not relevant when using the user interfaces provided by Cisco.

Requires user role: ADMIN, USER

Default value: 2

**USAGE:**

xConfiguration Video Presentation DefaultSource: *DefaultSource*

where

*DefaultSource*: 2

The video input source to use as default presentation source.

### xConfiguration Video Selfview Default Mode

Define 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 Selfview Default PIPPosition and the Video Selfview Default FullscreenMode settings respectively.

Requires user role: ADMIN

Default value: Current

**USAGE:**

xConfiguration Video Selfview Default Mode: *Mode*

where

*Mode*: 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.

### xConfiguration Video Selfview Default FullscreenMode

Define 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 Selfview Default Mode setting).

Requires user role: ADMIN

Default value: Current

#### USAGE:

xConfiguration Video Selfview Default FullscreenMode: FullscreenMode

where

FullscreenMode: 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.

### xConfiguration Video Selfview Default OnMonitorRole

Define which monitor to display the main video source (self-view) on after a call. Note that this video system has only one monitor, so this value is fixed and cannot be changed.

Requires user role: ADMIN

Default value: First

#### USAGE:

xConfiguration Video Selfview Default OnMonitorRole: OnMonitorRole

where

OnMonitorRole: First

First: The self-view picture will be shown on the integrated screen.

### xConfiguration Video Selfview Default PIPPosition

Define 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 Selfview Default Mode setting) and fullscreen view is switched off (see the Video Selfview Default FullscreenMode setting).

Requires user role: ADMIN

Default value: Current

#### USAGE:

xConfiguration Video Selfview Default PIPPosition: PIPPosition

where

PIPPosition: 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.

### xConfiguration Video Selfview OnCall Mode

This setting is used to switch on self-view for a short while when setting up a call. The Video Selfview OnCall Duration setting determines for how long it remains on. This applies when self-view in general is switched off.

Requires user role: ADMIN

Default value: On

#### USAGE:

xConfiguration Video Selfview OnCall Mode: Mode

where

Mode: Off/On

Off: self-view is not shown automatically during call setup.

On: self-view is shown automatically during call setup.

### xConfiguration Video Selfview OnCall Duration

This setting only has an effect when the Video Selfview OnCall 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

Default value: 10

#### USAGE:

xConfiguration Video Selfview OnCall Duration: Duration

where

Duration: Integer (1..60)

Range: Choose for how long self-view remains on. The valid range is between 1 and 60 seconds.



## Chapter 4

# xCommand commands



## Description of the xCommand commands

In this chapter, you can find a complete list of all xCommand type commands with parameters.

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

Go to: <http://www.cisco.com/go/dx-docs>

<b>Audio commands</b> .....	<b>76</b>	xCommand Call FarEndControl Camera Move .....	81
xCommand Audio Diagnostics MeasureDelay .....	76	xCommand Call FarEndControl Camera Stop.....	81
xCommand Audio Microphones Mute.....	76	xCommand Call FarEndControl RoomPreset Activate .....	81
xCommand Audio Microphones Unmute.....	76	xCommand Call FarEndControl Source Select .....	82
xCommand Audio SoundsAndAlerts Ringtone List .....	76	xCommand Call Forward.....	82
xCommand Audio SoundsAndAlerts Ringtone Play .....	76	xCommand Call Hold.....	82
xCommand Audio Select .....	77	xCommand Call Ignore.....	82
xCommand Audio Volume Decrease .....	77	xCommand Call Join .....	82
xCommand Audio Volume Increase .....	77	xCommand Call Reject.....	83
xCommand Audio Volume Set .....	77	xCommand Call Resume.....	83
xCommand Audio Volume SetToDefault .....	78	xCommand Call UnattendedTransfer .....	83
xCommand Audio Volume Mute.....	78		
xCommand Audio Volume Unmute .....	78	<b>CallHistory commands</b> .....	<b>83</b>
xCommand Audio VuMeter Start .....	78	xCommand CallHistory AcknowledgeAllMissedCalls .....	83
xCommand Audio VuMeter Stop.....	78	xCommand CallHistory AcknowledgeMissedCall .....	83
xCommand Audio VuMeter StopAll.....	79	xCommand CallHistory DeleteAll .....	83
		xCommand CallHistory DeleteEntry .....	84
<b>Bookings commands</b> .....	<b>79</b>	xCommand CallHistory Get.....	84
xCommand Bookings Clear.....	79	xCommand CallHistory Recents .....	85
xCommand Bookings Get .....	79		
xCommand Bookings List .....	80	<b>Conference commands</b> .....	<b>85</b>
		xCommand Conference DoNotDisturb Activate .....	85
<b>Call commands</b> .....	<b>80</b>	xCommand Conference DoNotDisturb Deactivate .....	85
xCommand Call Accept.....	80	xCommand Conference SpeakerLock Release .....	85
xCommand Call DTMFSend .....	80	xCommand Conference SpeakerLock Set .....	86
xCommand Call Disconnect.....	81		

<b>Diagnostics commands</b> .....	<b>86</b>	<b>Provisioning commands</b> .....	<b>96</b>
xCommand Diagnostics Run .....	86	xCommand Provisioning CompleteUpgrade .....	96
<b>Dial commands</b> .....	<b>87</b>	xCommand Provisioning PostponeUpgrade .....	96
xCommand Dial .....	87	xCommand Provisioning StartUpgrade .....	96
<b>HttpFeedback commands</b> .....	<b>87</b>	xCommand Provisioning ExtensionMobility Login .....	96
xCommand HttpFeedback Deregister .....	87	xCommand Provisioning ExtensionMobility Logout .....	96
xCommand HttpFeedback Register .....	88	<b>Proximity commands</b> .....	<b>97</b>
<b>Peripherals commands</b> .....	<b>88</b>	xCommand Proximity Services Activate .....	97
xCommand Peripherals Connect .....	88	xCommand Proximity Services Deactivate .....	97
xCommand Peripherals HeartBeat .....	89	<b>Security commands</b> .....	<b>97</b>
xCommand Peripherals List .....	89	xCommand Security Persistency .....	97
xCommand Peripherals Pairing DeviceDiscovery Start .....	89	xCommand Security Session List .....	97
xCommand Peripherals Pairing Pair .....	89	xCommand Security Session Terminate .....	98
xCommand Peripherals Pairing Unpair .....	90	<b>Standby commands</b> .....	<b>98</b>
xCommand Peripherals Purge .....	90	xCommand Standby Activate .....	98
<b>Phonebook commands</b> .....	<b>90</b>	xCommand Standby Deactivate .....	98
xCommand Phonebook Contact Add .....	90	xCommand Standby ResetTimer .....	98
xCommand Phonebook Contact Delete .....	91	<b>SystemUnit commands</b> .....	<b>99</b>
xCommand Phonebook Contact Modify .....	91	xCommand SystemUnit Boot .....	99
xCommand Phonebook ContactMethod Add .....	92	xCommand SystemUnit FactoryReset .....	99
xCommand Phonebook ContactMethod Delete .....	92	xCommand SystemUnit Notifications RemoveAll .....	99
xCommand Phonebook ContactMethod Modify .....	93	xCommand SystemUnit OptionKey Add .....	99
xCommand Phonebook Folder Add .....	93	xCommand SystemUnit OptionKey List .....	99
xCommand Phonebook Folder Delete .....	93	xCommand SystemUnit OptionKey Remove .....	99
xCommand Phonebook Folder Modify .....	94	xCommand SystemUnit OptionKey RemoveAll .....	100
xCommand Phonebook Search .....	94	xCommand SystemUnit SoftReset .....	100
<b>Presentation commands</b> .....	<b>95</b>	xCommand SystemUnit SoftwareUpgrade .....	100
xCommand Presentation Start .....	95	<b>Time commands</b> .....	<b>101</b>
xCommand Presentation Stop .....	95	xCommand Time DateTime Get .....	101
		xCommand Time DateTime Set .....	101



**UserInterface commands ..... 101**

- xCommand UserInterface Extensions Widget SetValue..... 101
- xCommand UserInterface Extensions Clear ..... 101
- xCommand UserInterface Extensions List ..... 101
- xCommand UserInterface Message Alert Clear ..... 102
- xCommand UserInterface Message Alert Display ..... 102
- xCommand UserInterface Message Prompt Clear ..... 102
- xCommand UserInterface Message Prompt Display ..... 103
- xCommand UserInterface Message Prompt Response ..... 103
- xCommand UserInterface Message TextLine Clear ..... 103
- xCommand UserInterface Message TextLine Display ..... 104
- xCommand UserInterface OSD Key Click ..... 104
- xCommand UserInterface OSD Key Press..... 104
- xCommand UserInterface OSD Key Release ..... 104
- xCommand UserInterface Presentation ExternalSource Add..... 105
- xCommand UserInterface Presentation ExternalSource Remove ..... 105
- xCommand UserInterface Presentation ExternalSource RemoveAll ..... 105
- xCommand UserInterface Presentation ExternalSource Select..... 105
- xCommand UserInterface Presentation ExternalSource State Set ..... 106

**UserManagement commands..... 106**

- xCommand UserManagement RemoteSupportUser Create ..... 106
- xCommand UserManagement RemoteSupportUser Delete..... 106
- xCommand UserManagement RemoteSupportUser DisablePermanently ..... 106
- xCommand UserManagement RemoteSupportUser GetState..... 107
- xCommand UserManagement User Passphrase Change ..... 107
- xCommand UserManagement User Passphrase Set ..... 107

**Video commands..... 108**

- xCommand Video ActiveSpeakerPIP Set..... 108
- xCommand Video Input SetMainVideoSource..... 108
- xCommand Video Layout LayoutFamily Set ..... 108
- xCommand Video PresentationPIP Set..... 108
- xCommand Video PresentationView Set ..... 109
- xCommand Video Selfview Set..... 109

## Audio commands

### xCommand Audio Diagnostics MeasureDelay

This command will measure the audio delay/latency in a device connected to the video system. A typical use case it to measure the delay in a TV connected to the video system via the HDMI connector. If the delay in a TV is too high, the real-time experience of a video call will be substantially degraded. If the delay is more than 50 ms we recommend the user to find a TV setting that has shorter delay. Typical TV settings that can reduce the delay are: Gaming Mode and PC Mode.

This command will measure the delay between output and input (microphone). This means that if you are using an external acoustic echo canceller or audio mixer, this command will also measure the delay of these devices.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Audio Diagnostics MeasureDelay [MeasurementLength: MeasurementLength] [Output: Output] [Volume: Volume]

where

MeasurementLength: Integer (1..100)

The length of the measurement in number of seconds. In a noisy environment a longer measurement time will give a more reliable and robust result.

Output: HDMI/Line/Internal/All

HDMI: Measure the delay of the device connected to the HDMI output.

Line: Measure the delay of the device connected to the Line output (only available on video systems that have a Line output).

Internal: Measure the delay of the internal loudspeaker (only available on video systems that have an internal loudspeaker).

All: Will measure the maximum delay of all outputs.

Volume: Integer (1..50)

The volume of the test signal. In a noisy environment a higher volume will give more reliable and robust results.

### xCommand Audio Microphones Mute

Mute all microphones.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Audio Microphones Mute

### xCommand Audio Microphones Unmute

Unmute all microphones.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Audio Microphones Unmute

### xCommand Audio SoundsAndAlerts Ringtone List

List all available ringtones. Use the xConfiguration Audio SoundsAndAlerts RingTone setting to choose a ringtone.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Audio SoundsAndAlerts Ringtone List

### xCommand Audio SoundsAndAlerts Ringtone Play

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

Requires user role: ADMIN, USER

#### USAGE:

xCommand Audio SoundsAndAlerts Ringtone Play RingTone: "RingTone"

where

RingTone: String (1, 100)

The name of the ringtone.

### xCommand Audio Select

Select which type of audio device to use (built-in loudspeakers and microphone, headsets, or handset).

Requires user role: ADMIN, USER

#### USAGE:

xCommand Audio Select Device: Device

where

Device: Internal/HeadsetAnalog/HeadsetUSB/HandsetUSB

Choose whether to use the built-in loudspeakers and microphone (internal), an analog headset (DX70 only), a USB headset, or a USB handset.

### xCommand Audio Volume Decrease

Decrease the volume on one of the endpoint's audio devices (built-in loudspeakers, headsets, or handset). By default, the volume is decreased by 5 steps (each step is 0.5 dB). Use the Steps parameter if you want to override the default behavior.

You can use the optional Device parameter to specify which device to address. The most recently selected device is chosen if you don't specify a device (see xStatus Audio SelectedDevice). Also refer to xCommand Audio Select.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Audio Volume Decrease [Device: Device] [Steps: Steps]

where

Device: Internal/HeadsetAnalog/HeadsetUSB/HandsetUSB

Choose whether to decrease the volume on the built-in loudspeakers (internal), an analog headset (DX70 only), a USB headset, or a USB handset.

Steps: Integer (1..10)

Choose the number of steps to decrease the volume with. One step is 0.5 dB.

### xCommand Audio Volume Increase

Increase the volume on one of the endpoint's audio devices (built-in loudspeakers, headsets, or handset). By default, the volume is increased by 5 steps (each step is 0.5 dB). Use the Steps parameter if you want to override the default behavior.

You can use the optional Device parameter to specify which device to address. The most recently selected device is chosen if you don't specify a device (see xStatus Audio SelectedDevice). Also refer to xCommand Audio Select.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Audio Volume Increase [Device: Device] [Steps: Steps]

where

Device: Internal/HeadsetAnalog/HeadsetUSB/HandsetUSB

Choose whether to increase the volume on the built-in loudspeakers (internal), an analog headset (DX70 only), a USB headset, or a USB handset.

Steps: Integer (1..10)

Choose the number of steps to increase the volume with. One step is 0.5 dB.

### xCommand Audio Volume Set

Set the volume on one of the endpoint's audio devices (built-in loudspeakers, headsets, or handset) to a specified level. You can use the optional Device parameter to specify which device to address. The most recently selected device is chosen if you don't specify a device (see xStatus Audio SelectedDevice). Also refer to xCommand Audio Select.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Audio Volume Set [Device: Device] Level: Level]

where

Device: Internal/HeadsetAnalog/HeadsetUSB/HandsetUSB

Choose whether to set the volume on the built-in loudspeakers (internal), an analog headset (DX70 only), a USB headset, or a USB handset.

Level: Integer (0..100)

Select the gain level. 70 equals 0 dB gain, and 100 equals 15 dB gain.

### xCommand Audio Volume SetToDefault

Set the volume on one of the endpoint's audio devices (built-in loudspeakers, headsets, or handset) to the default level as defined in the xConfiguration Audio DefaultVolume setting. You can use the optional Device parameter to specify which device to address. The most recently selected device is chosen if you don't specify a device (see xStatus Audio SelectedDevice). Also refer to xCommand Audio Select.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Audio Volume SetToDefault [Device: Device]

where

Device: Internal/HeadsetAnalog/HeadsetUSB/HandsetUSB

Choose whether to set the volume on the built-in loudspeakers (intern), an analog headset (DX70 only), a USB headset, or a USB handset.

### xCommand Audio Volume Mute

Mute the volume on the selected audio device (built-in loudspeakers, headsets, or handset). Refer to the xStatus Audio SelectedDevice and xCommand Audio Select commands for more information about the selected device.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Audio Volume Mute

### xCommand Audio Volume Unmute

Set the volume on the endpoint back on after muting.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Audio Volume Unmute

### xCommand Audio VuMeter Start

Start a VU meter to show the audio signal level on the specified connector. You have to specify both the connector's type and number (ConnectorType, ConnectorId) to uniquely identify the connector. The VU meter measures the input level for frequencies below 20 kHz.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Audio VuMeter Start ConnectorID: ConnectorID ConnectorType: ConnectorType

where

ConnectorID: Integer(1..1)

Select a connector number. Together with the ConnectorType it uniquely identifies the connector. Microphone [1] refers to the selected audio device (built-in loudspeakers, headsets, or handset). Refer to the xStatus Audio SelectedDevice and xCommand Audio Select commands for more information about the selected device.

ConnectorType: HDMI/Microphone

Select the connector type.

### xCommand Audio VuMeter Stop

Stop the VU meter on the specified connector. You have to specify both the connector's type and number (ConnectorType, ConnectorId) to uniquely identify the connector.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Audio VuMeter Stop ConnectorID: ConnectorID ConnectorType: ConnectorType

where

ConnectorID: Integer(1..1)

Select a connector number. Together with the ConnectorType it uniquely identifies the connector. Microphone [1] refers to the selected audio device (built-in loudspeakers, headsets, or handset). Refer to the xStatus Audio SelectedDevice and xCommand Audio Select commands for more information about the selected device.

ConnectorType: HDMI/Microphone

Select the connector type.

### xCommand Audio VuMeter StopAll

Stop all VU meters.

Requires user role: ADMIN, USER

#### USAGE:

```
xCommand Audio VuMeter StopAll
```

## Bookings commands

### xCommand Bookings Clear

Clear the current stored list of bookings.

Requires user role: ADMIN, USER

#### USAGE:

```
xCommand Bookings Clear
```

### xCommand Bookings Get

Get the booking information for a specific ID.

Requires user role: ADMIN, USER

#### USAGE:

```
xCommand Bookings Get Id: "Id"
```

where

Id: String (0, 128)

The booking ID of a call or conference.

## 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 set, past, present and future bookings are all listed. To avoid listing bookings from yesterday and before, use DayOffset = 0.

Requires user role: ADMIN, USER

### USAGE:

xCommand Bookings List [Days: Days] [DayOffset: DayOffset] [Limit: Limit] [Offset: Offset]

where

Days: Integer (1..365)

Number of days to retrieve bookings from.

DayOffset: Integer (0..365)

Which day to start the search from (today: 0, tomorrow: 1...).

Limit: Integer (1..65534)

Max number of bookings to list.

Offset: Integer (0..65534)

Offset number of bookings for this search.

## Call commands

### xCommand Call Accept

Accept an incoming call. If no CallId is specified, all incoming calls are accepted.

Requires user role: ADMIN, USER

### USAGE:

xCommand Call Accept [CallId: CallId]

where

CallId: Integer (0..65534)

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.

### xCommand Call DTMFSend

Send DTMF tones to the far end.

Requires user role: ADMIN, USER

### USAGE:

xCommand Call DTMFSend [CallId: CallId] DTMFString: "DTMFString"

where

CallId: Integer (0..65534)

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: String (0, 32)

Enter the DTMF string.



### xCommand Call Disconnect

Disconnect a call.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Call Disconnect [CallId: CallId]

where

CallId: Integer (0..65534)

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.

### xCommand Call 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.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Call FarEndControl Camera Move [CallId: CallId] Value: Value

where

CallId: Integer (0..65534)

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: Left/Right/Up/Down/ZoomIn/ZoomOut

Select the action for how to move the camera.

### xCommand Call FarEndControl Camera Stop

Stop the far end camera after the xCommand FarEndControl Camera Move has been issued.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Call FarEndControl Camera Stop [CallId: CallId]

where

CallId: Integer (0..65534)

The CallId is returned when the xCommand Dial command is run. During a call you can run the xStatus Call command to see the CallId.

### xCommand Call FarEndControl RoomPreset 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.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Call FarEndControl RoomPreset Activate [CallId: CallId] PresetId: PresetId

where

CallId: Integer (0..65534)

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 a call you can run the xStatus Call command to see the CallId.

PresetId: Integer (1..15)

The ID of the preset that is stored on the far end codec.

### xCommand Call FarEndControl Source Select

Select which video input source to use as the main source on the far end system.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Call FarEndControl Source Select [CallId: CallId] SourceId: SourceId

where

CallId: Integer (0..65534)

The CallID is returned when the xCommand Dial command is run. During a call you can run the xStatus Call command to see the CallId.

SourceId: Integer (0..15)

Select a video input source on the far end.

### xCommand Call Forward

Specifies what number or URI you want to forward your incoming calls to. The display name is a local reference for the forwarded destination. A message, together with the local reference, is shown on screen when you have configured the system to forward all calls.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Call Forward DisplayName: "DisplayName" Number: "Number"

where

DisplayName: String (0, 255)

The local reference for the forwarded destination.

Number: String (0, 255)

The directory number or URI which you want to forward the incoming calls to.

### xCommand Call Hold

Put a call on hold.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Call Hold [CallId: CallId] [Reason: Reason]

where

CallId: Integer (0..65534)

The CallID is returned when the xCommand Dial command is run. During a call you can run the xStatus Call command to see the CallId.

Reason: Conference/Transfer/Other

Internal usage only.

### xCommand Call Ignore

Turns off the ringtone for the incoming call. The call can still be answered.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Call Ignore CallId: CallId

where

CallId: Integer (0..65534)

The CallID is returned when the xCommand Dial command is run. During a call you can run the xStatus Call command to see the CallId.

### xCommand Call Join

Internal usage only.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Call Join CallId: CallId

where

CallId: Integer (0..65534)

### xCommand Call Reject

Reject incoming call. If no call id is specified, all incoming calls are rejected.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Call Reject [CallId: CallId]

where

CallId: Integer (0..65534)

The CallID is returned when the xCommand Dial command is run. During a call you can run the xStatus Call command to see the CallId.

### xCommand Call Resume

Resume a call that have been put on hold.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Call Resume [CallId: CallId]

where

CallId: Integer (0..65534)

The CallID is returned when the xCommand Dial command is run. During a call you can run the xStatus Call command to see the CallId.

### xCommand Call UnattendedTransfer

Transfers an ongoing call to another participant. Fully supported for SIP calls only.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Call UnattendedTransfer CallId: CallId Number: "Number"

where

CallId: Integer (0..65534)

The CallID is returned when the xCommand Dial command is run. During a call you can run the xStatus Call command to see the CallId.

Number: String (0, 255)

The number the call is transferred to.

## CallHistory commands

### xCommand CallHistory AcknowledgeAllMissedCalls

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

Requires user role: ADMIN, USER

#### USAGE:

xCommand CallHistory AcknowledgeAllMissedCalls

### xCommand CallHistory AcknowledgeMissedCall

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

Requires user role: ADMIN, USER

#### USAGE:

xCommand CallHistory AcknowledgeMissedCall CallHistoryId: CallHistoryId [AcknowledgeConsecutiveDuplicates: AcknowledgeConsecutiveDuplicates]

where

CallHistoryId: Integer (1..2147483647)

CallHistoryId for the call in question. Run xCommand CallHistory Get to get the id number.

AcknowledgeConsecutiveDuplicates: False/True

You can include or exclude all surrounding calls with duplicate information.

### xCommand CallHistory DeleteAll

Deletes all information on previous calls.

Requires user role: ADMIN, USER

#### USAGE:

xCommand CallHistory DeleteAll [Filter: Filter]

where

Filter: All/Missed/Placed/Received

You can filter which calls to delete.

### xCommand CallHistory DeleteEntry

Deletes all information on the specified call.

Requires user role: ADMIN, USER

#### USAGE:

xCommand CallHistory DeleteEntry CallHistoryId: CallHistoryId  
 [AcknowledgeConsecutiveDuplicates: AcknowledgeConsecutiveDuplicates]

where

CallHistoryId: Integer (1..2147483647)

CallHistoryId for the call in question. Run xCommand CallHistory Get to get the id number.

AcknowledgeConsecutiveDuplicates: False/True

You can include or exclude all surrounding calls with duplicate information.

### xCommand CallHistory Get

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

Requires user role: ADMIN, USER

#### USAGE:

xCommand CallHistory Get [Filter: Filter] [Offset: Offset] [Limit: Limit]  
 [DetailLevel: DetailLevel] [SearchString: "SearchString"] [CallHistoryId:  
CallHistoryId]

where

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

You can filter which calls to retrieve.

Offset: Integer (0..65534)

Sets the call from which to start.

Limit: Integer (0..65534)

Defines the amount of calls in the output.

DetailLevel: Basic/Full

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

SearchString: String ( 0, 255)

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

CallHistoryId: Integer (0..65534)

CallHistoryId for the call in question.

## xCommand CallHistory Recents

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

Requires user role: ADMIN, USER

### USAGE:

```
xCommand CallHistory Recents [Filter: Filter] [Offset: Offset] [Limit: Limit]
[DetailLevel: DetailLevel] [SearchString: "SearchString"] [CallHistoryId:
CallHistoryId] [Order: Order]
```

where

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

You can filter which calls to retrieve.

*Offset*: Integer (0..65534)

Sets the call from which to start.

*Limit*: Integer (0..65534)

Defines the amount of calls in the output.

*DetailLevel*: Basic/Full

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

*SearchString*: String (0, 255)

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

*CallHistoryId*: Integer (0..65534)

CallHistoryId for the call in question.

*Order*: OccurrenceTime/OccurrenceFrequency

Define the order in which the previous calls are presented.

## Conference commands

### xCommand Conference DoNotDisturb Activate

This command switches on the Do Not Disturb mode, and the Timeout parameter allows you to control when it is switched off again. When Do Not Disturb is switched on, all incoming calls are rejected and registered as missed calls. The calling side receives a busy signal.

Requires user role: ADMIN, USER

### USAGE:

```
xCommand Conference DoNotDisturb Activate [Timeout: Timeout]
```

where

*Timeout*: Integer (1..1440)

Set the number of minutes before Do Not Disturb is switched off. If not set, Do Not Disturb times out after 1440 minutes (24 hours).

### xCommand Conference DoNotDisturb Deactivate

Switch off the Do Not Disturb mode. When Do Not Disturb is switched off incoming calls come through as normal.

Requires user role: ADMIN, USER

### USAGE:

```
xCommand Conference DoNotDisturb Deactivate
```

### xCommand Conference SpeakerLock Release

Releases locked speaker set by xCommand Conference SpeakerLock Set. Default voice switching is switched back on.

Requires user role: ADMIN, USER

### USAGE:

```
xCommand Conference SpeakerLock Release
```

### xCommand Conference SpeakerLock Set

For manually locking one of the speakers to the prominent speaker position. This overrides the default voice switching.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Conference SpeakerLock Set Target: Target [CallId: CallId]

where

Target: local/remote

Identifies local or remote participant.

CallId: Integer (0..65534)

Identify CallID for the remote participant. Only relevant if Target is set to "remote".

## Diagnostics commands

### xCommand Diagnostics Run

This command runs self-diagnostics commands on the system.

Requires user role: ADMIN

#### USAGE:

xCommand Diagnostics Run [ResultSet: ResultSet]

where

ResultSet: Alerts/All/None

You can filter the diagnostics results to alerts, all or none. If not set, the result will show all results.

## Dial commands

### xCommand Dial

Dial out from the system. Returns information about the CallId and Conferenceld, which are required for some of the other commands.

Requires user role: ADMIN, USER

#### USAGE:

```
xCommand Dial Number: "Number" [Protocol: Protocol] [CallRate: CallRate]
[CallType: CallType] [BookingId: "BookingId"] [Appearance: Appearance]
[DisplayName: "DisplayName"]
```

where

Number: String (0, 255)

Enter the number or address.

Protocol: H320/H323/Sip/Spark

Select the call protocol.

CallRate: Integer (64..6000)

Set the call rate.

CallType: Audio/Video

Select the call type.

BookingId: String (0, 255)

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.

Appearance: Integer (1..999999999)

Internal usage only.

DisplayName: String (0, 255)

The display name of the remote participant.

## HttpFeedback commands

### xCommand HttpFeedback Deregister

Deregister the HTTP feedback over HTTP(S).

Requires user role: ADMIN

#### USAGE:

```
xCommand HttpFeedback Deregister FeedbackSlot: FeedbackSlot
```

where

FeedbackSlot: Integer (1..4)

Deregister the feedback slot that was registered using the xCommand HttpFeedback Register command.

## xCommand HttpFeedback Register

Register the system to a HTTP(S) server to return XML feedback over HTTP(S) to specific URLs.

Requires user role: ADMIN

### USAGE:

```
xCommand HttpFeedback Register FeedbackSlot: FeedbackSlot [Format: Format]
ServerUrl: "ServerUrl" [Expression[1..15]: "Expression[1..15]"]
```

where

FeedbackSlot: Integer (1..4)

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.

Format: XML/JSON

Set the format for the feedback from the HTTP server to XML or JSON.

ServerUrl: String (1, 2048)

The URL to the HTTP server where you want the codec to post the HTTP feedback messages to.

Expression[1..15]: String (1, 255)

The XPath expression specifies which parts of the Status, Configuration or Event XML documents are monitored. You can have from 1 to 15 XPath expressions.

## 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.

Requires user role: ADMIN, USER

### USAGE:

```
xCommand Peripherals Connect [HardwareInfo: "HardwareInfo"] ID: "ID" [Name:
"Name"] [NetworkAddress: "NetworkAddress"] [SerialNumber: "SerialNumber"]
[SoftwareInfo: "SoftwareInfo"] Type: Type
```

where

HardwareInfo: String (0, 100)

The device's hardware number.

ID: String (1, 100)

A unique ID for the device you are connecting to, typically a MAC address.

Name: String (0, 100)

Define a name for the device.

NetworkAddress: String (0, 100)

Network address for the device you are connecting to.

SerialNumber: String (0, 100)

The device's serial number.

SoftwareInfo: String (0, 100)

Software version the device is running.

Type: Byod/ControlSystem/Other/TouchPanel

Define the type of device you are connecting to.



### 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 to cameras.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Peripherals HeartBeat ID: "ID" [Timeout: Timeout]

where

ID: String (1, 100)

A unique ID for the device you are connecting to, typically a MAC address.

Timeout: Integer (1..65535)

Set how long the device will send heartbeat.

### xCommand Peripherals List

Lists all currently and previously connected peripherals.

Requires user role: ADMIN, ROOMCONTROL

#### USAGE:

xCommand Peripherals List [Connected: Connected] [Type: Type]

where

Connected: False/True

Limit the search to currently connected devices.

Type: All/ControlSystem/ISDNLink/Other/TouchPanel

Limit the search by device type.

### xCommand Peripherals Pairing DeviceDiscovery Start

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

Requires user role: ADMIN

#### USAGE:

xCommand Peripherals Pairing DeviceDiscovery Start [AutoPairing: AutoPairing]  
[DeviceType: DeviceType] [Timeout: Timeout]

where

AutoPairing: On/Off

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

DeviceType: ISDNLink

Only look for ISDN Link.

Timeout: Integer (3..60)

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

### xCommand Peripherals Pairing Pair

Pair an ISDN Link to an endpoint.

Requires user role: ADMIN

#### USAGE:

xCommand Peripherals Pairing Pair MacAddress: "MacAddress"

where

MacAddress: String (1, 1450)

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

### xCommand Peripherals Pairing Unpair

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

Requires user role: ADMIN

#### USAGE:

xCommand Peripherals Pairing Unpair MacAddress: "MacAddress"

where

MacAddress: String (1, 100)

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

### xCommand Peripherals 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, USER

#### USAGE:

xCommand Peripherals Purge ID: "ID"

where

ID: String (1, 100)

Mac address of the ISDN Link in the format "xx:xx:xx:xx:xx:xx".

## Phonebook commands

### xCommand Phonebook Contact Add

Requires user role: ADMIN, USER

#### USAGE:

xCommand Phonebook Contact Add Name: "Name" [FolderId: "FolderId"] [ImageURL: "ImageURL"] [Title: "Title"] [Number: "Number"] [Protocol: "Protocol"] [CallRate: "CallRate"] [CallType: CallType] [Device: Device] [Tag: Tag]

where

Name: String (0, 255)

The name of the contact.

FolderId: String (0, 255)

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: String (0, 255)

Currently not in use.

Title: String (0, 255)

The title of the contact.

Number: String (0, 255)

The phone number or address of the contact.

Protocol: Auto/H320/H323/SIP/Spark

Select the Auto, Spark, SIP, H323 or H320 protocol.

CallRate: Integer (0..6000)

Set a call rate.

CallType: Audio/Video

Select a call type (audio or video).

Device: Mobile/Other/Telephone/Video

Select the device type.

Tag: Untagged/Favorite

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

### xCommand Phonebook Contact Delete

Delete an existing contact from the local phonebook.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Phonebook Contact Delete ContactId: "ContactId"

where

ContactId: String (0, 255)

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.

### 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.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Phonebook Contact Modify ContactId: "ContactId" [Name: "Name"] [FolderId: "FolderId"] [ImageURL: "ImageURL"] [Title: "Title"] [Tag: "Tag"]

where

ContactId: String (0, 255)

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: String (0, 255)

The name of the contact.

FolderId: String (0, 255)

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: String (0, 255)

Currently not in use.

Title: String (0, 255)

The title of the contact.

Tag: Untagged/Favorite

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

### xCommand Phonebook ContactMethod Add

Add contact details for an existing contact in the local phonebook. The command returns the ContactMethodId, which is a unique string that identifies the contact method; typically the format is "n".

You can add several contact methods to a contact. Note that only the first contact method will appear in the Favorites list on the Cisco TelePresence Touch controller. The first contact method may have been created when issuing the xCommand Phonebook Contact Add command to make the contact. All contact methods are available in the API, on the web interface and when using the remote control.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Phonebook ContactMethod Add ContactId: "ContactId" [Device: Device]  
 Number: "Number" [Protocol: Protocol] [CallRate: CallRate] [CallType: CallType]

where

ContactId: String (0, 255)

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: Mobile/Other/Telephone/Video

Set which type of device to call to.

Number: String (0, 255)

The phone number or address of the contact.

Protocol: Auto/H320/H323/SIP/Spark

Select protocol.

CallRate: Integer (0..6000)

Set a call rate.

CallType: Audio/Video

Select a call type (audio or video).

### xCommand Phonebook ContactMethod Delete

Delete a contact method from an existing contact in the local phonebook.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Phonebook ContactMethod Delete ContactId: "ContactId" ContactMethodId: "ContactMethodId"

where

ContactId: String (0, 255)

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: String (0, 255)

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.

### xCommand Phonebook ContactMethod Modify

Modify details about the contact method for an existing contact in the local phonebook.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Phonebook ContactMethod Modify ContactId: "ContactId" ContactMethodId: "ContactMethodId" [Device: Device] [Number: "Number"] [Protocol: Protocol] [CallRate: CallRate] [CallType: CallType]

where

ContactId: String (0, 255)

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: String (0, 255)

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: Mobile/Other/Telephone/Video

Set which type of device to call to.

Number: String (0, 255)

The phone number or address of the contact.

Protocol: Auto/H320/H323/SIP/Spark

Select protocol.

CallRate: Integer (0..6000)

Set a call rate.

CallType: Audio/Video

Select a call type (audio or video).

### 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".

Requires user role: ADMIN, USER

#### USAGE:

xCommand Phonebook Folder Add Name: "Name" [ParentFolderId: "ParentFolderId"]

where

Name: String (0, 255)

The name of the folder.

ParentFolderId: String (0, 255)

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.

### xCommand Phonebook Folder Delete

Delete an existing folder from the local phonebook.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Phonebook Folder Delete FolderId: "FolderId"

where

FolderId: String (0, 255)

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.

## xCommand Phonebook Folder Modify

Modify an existing phonebook folder.

Requires user role: ADMIN, USER

### USAGE:

```
xCommand Phonebook Folder Modify FolderId: "FolderId" [Name: "Name"]
[ParentFolderId: "ParentFolderId"]
```

where

FolderId: String (0, 255)

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: String (0, 255)

The name of the contact.

ParentFolderId: String (0, 255)

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.

## 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.

Requires user role: ADMIN, USER

### USAGE:

```
xCommand Phonebook Search [PhonebookId: "PhonebookId"] [PhonebookType:
PhonebookType] [SearchString: "SearchString"] [SearchField: SearchField] [Offset:
Offset] [FolderId: "FolderId"] [Limit: Limit] [Recursive: Recursive] [ContactType:
ContactType] [Tag: Tag]
```

where

PhonebookId: String (0, 255)

The identifier of the phonebook server that will be searched. See the xConfiguration Phonebook Server ID setting.

PhonebookType: Corporate/Local

Define whether to search the local phone book or the corporate phonebook.

SearchString: String (0, 255)

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: Name/Number

Currently not in use.

Offset: Integer (0..65534)

Get records starting with this offset in a search. The default value is 0. Offset is used together with Limit to support paging.

FolderId: String (0, 255)

Search only in the specified folder. The FolderId (string) is listed in the ResultSet of a search result containing folders.

Limit: Integer (0..65534)

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: False/True

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.

ContactType: Any/Folder/Contact

Search all contact types, or limit the search to folders or individual contacts.

Tag: Untagged/Favorite

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

## Presentation commands

### xCommand Presentation Start

Open a media stream from the selected presentation source.

Requires user role: ADMIN, USER

**USAGE:**

xCommand Presentation Start [PresentationSource: PresentationSource] [SendingMode: SendingMode] [ConnectorId: ConnectorId] [Instance: Instance] where

PresentationSource: Integer (1..2)

Select the video input source to be used for presentation, identified by source number.

SendingMode: LocalRemote/LocalOnly

Select whether the presentation is shown locally or locally and remotely.

ConnectorId: Integer (1..2)

Select the video input source to be used for presentation, identified by connectorId.

Instance: New/1/2/3/4/5/6

Select which local presentation instance you wish to start.

### xCommand Presentation Stop

Stop the media stream from the presentation source.

Requires user role: ADMIN, USER

**USAGE:**

xCommand Presentation Stop [Instance: Instance] [PresentationSource: PresentationSource] where

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

Select which local presentation you wish to stop, identified by presentation instance.

PresentationSource: Integer (1..2)

Select which local presentation you wish to stop, identified by source number.

## Provisioning commands

### xCommand Provisioning CompleteUpgrade

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

Requires user role: ADMIN, USER

**USAGE:**

xCommand Provisioning CompleteUpgrade

### xCommand Provisioning PostponeUpgrade

Postpones the installing of the software upgrade.

Requires user role: ADMIN, USER

**USAGE:**

xCommand Provisioning PostponeUpgrade SecondsToPostpone: SecondsToPostpone

where

SecondsToPostpone: Integer (0..65534)

Set how long to postpone the upgrade. The value is in seconds.

### 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, USER

**USAGE:**

xCommand Provisioning StartUpgrade

### xCommand Provisioning ExtensionMobility Login

Login command for the Extension Mobility service. You log in to the Extension Mobility service with a user name (UserId) and pin code (Pin). The user name and pin code are set up in CUCM. CUCM also supports multiple profiles for a user.

If you, for a user that has multiple profiles, submit a login command with only user name and pin code, CUCM will send a list of available profiles back to the video system. Then the video system will create corresponding ExtensionMobilityProfileSelection Profile events, so that a new login command, which contains the Profile parameter in addition to the user name and pin code, can be submitted.

Requires user role: ADMIN, USER

**USAGE:**

xCommand Provisioning ExtensionMobility Login UserId: "UserId" Pin: "Pin"  
[Profile: "Profile"]

where

UserId: String (1, 255)

A valid user name, as set up in CUCM.

Pin: String (1, 255)

A valid pin code for the user, as set up in CUCM.

Profile: String (1, 255)

A valid profile for the user, as set up in CUCM. This parameter applies only if the user has multiple profiles.

### xCommand Provisioning ExtensionMobility Logout

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

Requires user role: ADMIN, USER

**USAGE:**

xCommand Provisioning ExtensionMobility Logout



## Proximity commands

### xCommand Proximity Services Activate

Reactivate the Proximity services that were deactivated with xCommand Proximity Services Deactivate.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Proximity Services Activate

### xCommand Proximity Services Deactivate

This command deactivates all proximity services on the endpoint. To reactivate proximity services use the command xCommand Proximity Services Activate.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Proximity Services Deactivate

## Security commands

### 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.

Requires user role: ADMIN

#### USAGE:

xCommand Security Persistency Configurations: Configurations CallHistory: CallHistory InternalLogging: InternalLogging LocalPhonebook: LocalPhonebook DHCP: DHCP ConfirmAndReboot: ConfirmAndReboot

where

Configurations: NonPersistent/Persistent

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

CallHistory: NonPersistent/Persistent

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

InternalLogging: NonPersistent/Persistent

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

LocalPhonebook: NonPersistent/Persistent

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

DHCP: NonPersistent/Persistent

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

ConfirmAndReboot: Yes

Reboots the system.

### xCommand Security Session List

List active sessions.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Security Session List

## xCommand Security Session Terminate

Terminate a session.

Requires user role: ADMIN, USER

### USAGE:

xCommand Security Session Terminate SessionId: "SessionId"

where

SessionId: String (0, 32)

The session ID number.

## 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: ADMIN, USER

### USAGE:

xCommand Standby Activate

### xCommand Standby Deactivate

Bring the system out of standby mode.

Requires user role: ADMIN, USER

### USAGE:

xCommand Standby Deactivate

### xCommand Standby ResetTimer

Set a temporary standby delay. 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: ADMIN, USER

### USAGE:

xCommand Standby ResetTimer Delay: Delay

where

Delay: Integer (1..480)

Set the delay in minutes.

## SystemUnit commands

### xCommand SystemUnit Boot

Reboot the system.

Requires user role: ADMIN, USER

#### USAGE:

`xCommand SystemUnit Boot [Action: Action]`

where

Action: Restart/Shutdown

As a default the system restarts after a reboot. By selecting Shutdown, the system will not restart.

### 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 behaviour can be forced by selecting a different TrailingAction.

Requires user role: ADMIN, USER

#### USAGE:

`xCommand SystemUnit FactoryReset Confirm: Confirm [TrailingAction: TrailingAction]`

where

Confirm: Yes

Include to confirm your choice.

TrailingAction: NoAction/Restart/Shutdown

Select Shutdown or NoAction to override the default behaviour (Restart).

### xCommand SystemUnit Notifications RemoveAll

Clears the list of system notifications that are reported by xStatus SystemUnit Notifications Text/Type.

Requires user role: ADMIN

#### USAGE:

`xCommand SystemUnit Notifications RemoveAll`

### xCommand SystemUnit OptionKey Add

Add an option key to support additional features.

Requires user role: ADMIN

#### USAGE:

`xCommand SystemUnit OptionKey Add Key: "Key"`

where

Key: String (16, 24)

The key you have received for the option you wish to switch on.

### xCommand SystemUnit OptionKey List

List all option keys.

Requires user role: ADMIN

#### USAGE:

`xCommand SystemUnit OptionKey List`

### xCommand SystemUnit OptionKey Remove

Remove a specified option key.

Requires user role: ADMIN

#### USAGE:

`xCommand SystemUnit OptionKey Remove Type: Type`

where

Type: RemoteMonitoring

### xCommand SystemUnit OptionKey RemoveAll

Remove all option keys.

Requires user role: ADMIN

#### USAGE:

xCommand SystemUnit OptionKey RemoveAll Confirm: Confirm

where

Confirm: Yes

### xCommand SystemUnit SoftReset

Reset most parameters to their default values. This does not include parameters associated with room setup, such as camera position, language, and volume.

Requires user role: ADMIN, USER

#### USAGE:

xCommand SystemUnit SoftReset Confirm: Confirm

where

Confirm: Yes

### xCommand SystemUnit SoftwareUpgrade

Initiate a software upgrade by fetching the software from a given URL. If the server requires username and password these parameters must be included.

Requires user role: ADMIN

#### USAGE:

xCommand SystemUnit SoftwareUpgrade URL: "URL" [UserName: "UserName"] [Password: "Password"] [Forced: Forced]

where

URL: String (0, 255)

The software package location

UserName: String (0, 255)

User name to access the server location, if needed.

Password: String (0, 255)

Password to access the server location, if needed.

Forced: False/True

You can force a system to accept a software upgrade immediately, without giving users a chance to acknowledge or postpone the update.

## Time commands

### xCommand Time DateTime Get

Read the time and date from the system.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Time DateTime Get

### xCommand Time DateTime Set

Set the date and time for the system, if not available from NTP (Network Time Protocol).

Requires user role: ADMIN, USER

#### USAGE:

xCommand Time DateTime Set [Year: Year] [Month: Month] [Day: Day] [Hour: Hour] [Minute: Minute] [Second: Second]

where

Year: Integer (2015..2037)

Month: Integer (1..12)

Day: Integer (1..31)

Hour: Integer (0..23)

Minute: Integer (0..59)

Second: Integer (0..59)

## UserInterface commands

### xCommand UserInterface Extensions Widget SetValue

Set the value of the given widget. Updates the UserInterface Extensions status tree. Returns an error if the value is out of range.

Requires user role: ADMIN

#### USAGE:

xCommand UserInterface Extensions Widget SetValue Value: "Value" WidgetId: "WidgetId"

where

Value: String (0, 255)

The value of the widget. The range of values depends on the widget type.

WidgetId: String (0, 40)

The unique identifier for the widget.

### xCommand UserInterface Extensions Clear

Delete all user interface extensions (widgets) from the video system.

Requires user role: ADMIN

#### USAGE:

xCommand UserInterface Extensions Clear

### xCommand UserInterface Extensions List

List all user interface extensions (widgets) that exist on the video system.

Requires user role: ADMIN

#### USAGE:

xCommand UserInterface Extensions List

### xCommand UserInterface 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, USER, ROOMCONTROL

#### USAGE:

xCommand UserInterface Message Alert Clear

### xCommand UserInterface 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.

Requires user role: ADMIN, ROOMCONTROL

#### USAGE:

xCommand UserInterface Message Alert Display [Title: "Title"] Text: "Text"  
[Duration: Duration]

where

Title: String (0, 255)

Enter a message title.

Text: String (0, 255)

Enter the message to be displayed. The <p> and <br> HTML tags will result in line breaks as normal; any other tags will appear as plain text.

Duration: Integer (0..3600)

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.

### xCommand UserInterface 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.

Requires user role: ADMIN, USER, ROOMCONTROL

#### USAGE:

xCommand UserInterface Message Prompt Clear [FeedbackId: "FeedbackId"]

where

FeedbackId: String (0, 255)

The FeedbackId corresponds to the FeedbackId given by the xCommand Message Prompt Display command.

### xCommand UserInterface 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:

Text:

FeedbackId:

Option.1 to Option.5:

Requires user role: ADMIN, ROOMCONTROL

#### USAGE:

```
xCommand UserInterface Message Prompt Display [Title: "Title"] Text: "Text"
[FeedbackId: "FeedbackId"] [Option.1: "Option.1"] [Option.2: "Option.2"] [Option.3:
"Option.3"] [Option.4: "Option.4"] [Option.5: "Option.5"]
```

where

Title: String (0, 255)

Enter the message title.

Text: String (0, 255)

Enter the text line to be displayed. The <p> and <br> HTML tags will result in line breaks as normal; any other tags will appear as plain text.

FeedbackId: String (0, 255)

To identify the feedback enter a FeedbackId.

Option.1: String (0, 255)

Enter the text to appear on the feedback options.

Option.2: String (0, 255)

Enter the text to appear on the feedback options.

Option.3: String (0, 255)

Enter the text to appear on the feedback options.

Option.4: String (0, 255)

Enter the text to appear on the feedback options.

Option.5: String (0, 255)

Enter the text to appear on the feedback options.

### xCommand UserInterface Message Prompt Response

Give a response to the xCommand Message Prompt Display.

Use the xFeedback commands to monitor the feedback from the user. Read more about the xFeedback commands in the API introduction section in this guide.

Requires user role: ADMIN, USER, ROOMCONTROL

#### USAGE:

```
xCommand UserInterface Message Prompt Response [FeedbackId: "FeedbackId"]
OptionId: OptionId
```

where

FeedbackId: String (0, 255)

The FeedbackId corresponds to the FeedbackId given by the xCommand Message Prompt Display command.

OptionId: Integer (1..5)

The OptionId corresponds to the OptionIds given as possible responses in the xCommand Message Prompt Display command.

### xCommand UserInterface Message TextLine Clear

Clears the text line which was defined by the xCommand Message TextLine Display command.

Requires user role: ADMIN, USER, ROOMCONTROL

#### USAGE:

```
xCommand UserInterface Message TextLine Clear
```

### xCommand UserInterface 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.

Requires user role: ADMIN, ROOMCONTROL

#### USAGE:

xCommand UserInterface Message TextLine Display Text: "Text" [X: X] [Y: Y]  
[Duration: Duration]

where

Text: String (0, 140)

Enter the text line to be displayed. The <p> and <br> HTML tags will result in line breaks as normal; any other tags will appear as plain text.

X: Integer (1..10000)

Enter the X-coordinate (horizontal) on screen. X=0 is in the upper left corner.

Y: Integer (1..10000)

Enter the Y-coordinate (vertical) on screen. Y=0 is in the upper left corner.

Duration: Integer (0..3600)

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.

### xCommand UserInterface OSD Key Click

Emulates a remote control key press, followed by a key release.

Requires user role: ADMIN

#### USAGE:

xCommand UserInterface OSD Key Click Key: Key

where

Key: 0/1/2/3/4/5/6/7/8/9/C/Call/Disconnect/Down/F1/F2/F3/F4/F5/Grab/Home/Layout/Left/Mute/MuteMic/Ok/PhoneBook/Presentation/Right/Selfview/Square/SrcAux/SrcCamera/SrcDocCam/SrcPc/SrcVcr/Star/Up/VolumeDown/VolumeUp/ZoomIn/ZoomOut/+/-

Define the remote key to press.

### xCommand UserInterface OSD Key Press

Emulates a remote control key press without releasing it. The Key Press command must be followed

by a Key Release command to emulate releasing the key.

Requires user role: ADMIN

#### USAGE:

xCommand UserInterface OSD Key Press Key: Key

where

Key: 0/1/2/3/4/5/6/7/8/9/C/Call/Disconnect/Down/F1/F2/F3/F4/F5/Grab/Home/Layout/Left/Mute/MuteMic/Ok/PhoneBook/Presentation/Right/Selfview/Square/SrcAux/SrcCamera/SrcDocCam/SrcPc/SrcVcr/Star/Up/VolumeDown/VolumeUp/ZoomIn/ZoomOut/+/-

Define the remote key to press.

### xCommand UserInterface OSD Key Release

Emulates a remote control key release. The Key Release command is issued after a Key Press command.

Requires user role: ADMIN

#### USAGE:

xCommand UserInterface OSD Key Release Key: Key

where

Key: 0/1/2/3/4/5/6/7/8/9/C/Call/Disconnect/Down/F1/F2/F3/F4/F5/Grab/Home/Layout/Left/Mute/MuteMic/Ok/PhoneBook/Presentation/Right/Selfview/Square/SrcAux/SrcCamera/SrcDocCam/SrcPc/SrcVcr/Star/Up/VolumeDown/VolumeUp/ZoomIn/ZoomOut/+/-

Define the remote control key to release.



### xCommand UserInterface Presentation ExternalSource Add

Establish and set up an input source that is connected to the video system via an external switch.

Requires user role: ADMIN, ROOMCONTROL

**USAGE:**

xCommand UserInterface Presentation ExternalSource Add ConnectorId: ConnectorId  
 SourceIdentifier: "SourceIdentifier" Name: "Name" Type: Type

where

ConnectorId: Integer (2..2)

The ID of the video system connector that the external switch is connected to.

SourceIdentifier: String (0, 40)

A unique string that identifies the input source. The identifier is used by other commands and events when addressing this specific input source.

Name: String (0, 40)

The name of the input source. This name will be displayed on the user interface.

Type: camera/desktop/document\_camera/mediaplayer/PC/whiteboard/other

The type of input source. The type determines which icon to display on the user interface.

### xCommand UserInterface Presentation ExternalSource Remove

Remove the input source (specified by the SourceIdentifier) from the list of external input sources.

Requires user role: ADMIN, ROOMCONTROL

**USAGE:**

xCommand UserInterface Presentation ExternalSource Remove SourceIdentifier:  
 "SourceIdentifier"

where

SourceIdentifier: String (0, 40)

The unique string that identifies this input source. It was defined when the source was created with the UserInterface Presentation ExternalSource Add command.

### xCommand UserInterface Presentation ExternalSource RemoveAll

Remove all input sources from the list of external input sources.

Requires user role: ADMIN, ROOMCONTROL

**USAGE:**

xCommand UserInterface Presentation ExternalSource RemoveAll

### xCommand UserInterface Presentation ExternalSource Select

Starts to present the input source (specified by the SourceIdentifier) if it is in Ready state (see the UserInterface Presentation ExternalSource State Set command). The input source will be shown in the user interface sharetray as "Presenting".

Requires user role: ADMIN, USER, ROOMCONTROL

**USAGE:**

xCommand UserInterface Presentation ExternalSource Select SourceIdentifier:  
 "SourceIdentifier"

where

SourceIdentifier: String (0, 40)

The unique string that identifies this input source. It was defined when the source was created with the UserInterface Presentation ExternalSource Add command.

### xCommand UserInterface Presentation ExternalSource State Set

Set or change the state of the input source (specified by the SourceIdentifier).

Requires user role: ADMIN, ROOMCONTROL

#### USAGE:

```
xCommand UserInterface Presentation ExternalSource State Set SourceIdentifier:
"SourceIdentifier" State: State [ErrorReason: "ErrorReason"]
```

where

SourceIdentifier: String (0, 40)

The unique string that identifies this input source. It was defined when the source was created with the UserInterface Presentation ExternalSource Add command.

State: Hidden/Ready/NotReady/Error

Ready: The input source will be presented when someone selects it (taps it on the user interface, or when the UserInterface Presentation ExternalSource Select command is issued).

NotReady: The input source is not ready, and cannot be selected.

Error: The input source is in an erroneous state. The ErrorReason parameter may be used to inform the user about the reason for failure.

Hidden: The input source will not be displayed in the user interface sharetray.

ErrorReason: String (0, 80)

This string is displayed in the user interface sharetray, given that the State parameter is set to Error.

## 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.

Requires user role: ADMIN

#### USAGE:

```
xCommand UserManagement RemoteSupportUser Create [ExpiryDays: ExpiryDays]
```

where

ExpiryDays: Integer (1..31)

Define the duration for the passphrase validity. Default is 7 days.

### xCommand UserManagement RemoteSupportUser Delete

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

Requires user role: ADMIN

#### USAGE:

```
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

#### USAGE:

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

where

Confirm: Yes

### xCommand UserManagement RemoteSupportUser GetState

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

Requires user role: ADMIN

#### USAGE:

```
xCommand UserManagement RemoteSupportUser GetState
```

### xCommand UserManagement User Passphrase Change

Change the passphrase for the user you logged in as. If you are logged in as the administrator, this will change the administrator passphrase.

Requires user role: ADMIN, USER

#### USAGE:

```
xCommand UserManagement User Passphrase Change NewPassphrase: "NewPassphrase"
OldPassphrase: "OldPassphrase"
```

where

NewPassphrase: String (0, 255)

OldPassphrase: String (0, 255)

### xCommand UserManagement User Passphrase Set

Set a user passphrase for the specified user. You must be logged in as an administrator to set a user passphrase.

Requires user role: ADMIN

#### USAGE:

```
xCommand UserManagement User Passphrase Set NewPassphrase: "NewPassphrase"
Username: "Username" YourPassphrase: "YourPassphrase"
```

where

NewPassphrase: String (0, 255)

Username: String (0, 127)

YourPassphrase: String (0, 255)

## Video commands

### xCommand Video ActiveSpeakerPIP Set

Sets position for the active speakers PiP (picture in picture).

Requires user role: ADMIN, USER

#### USAGE:

xCommand Video ActiveSpeakerPIP Set Position: Position

where

Position: CenterLeft/CenterRight/LowerLeft/LowerRight/UpperCenter/UpperLeft/UpperRight

Select one of the predefined positions.

### 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.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Video Input SetMainVideoSource [ConnectorId: ConnectorId] [SourceId: SourceId]

where

ConnectorId: Integer (1..5)

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: Integer (1..4)

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.

### xCommand Video Layout LayoutFamily Set

Select the screen layout mode.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Video Layout LayoutFamily Set [Target: Target] [CallId: CallId] LayoutFamily: LayoutFamily [CustomLayoutName: "CustomLayoutName"]

where

Target: local/remote

Select if the target is the local layout or the remote layout.

CallId: Integer (0..65534)

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: auto/custom/equal/overlay/prominent/single

Select a layout family.

CustomLayoutName: String (1, 128)

Enter a name for the layout.

### xCommand Video PresentationPIP Set

Sets position for the presentation PiP (picture in picture).

Requires user role: ADMIN, USER

#### USAGE:

xCommand Video PresentationPIP Set Position: Position

where

Position: CenterLeft/CenterRight/LowerLeft/LowerRight/UpperCenter/UpperLeft/UpperRight

Select one of the predefined positions.

### xCommand Video PresentationView Set

Set the presentation view mode

Requires user role: ADMIN, USER

#### USAGE:

xCommand Video PresentationView Set View: View

where

View: Default/Maximized/Minimized

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.

### 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.

Requires user role: ADMIN, USER

#### USAGE:

xCommand Video Selfview Set [Mode: Mode] [FullscreenMode: FullscreenMode]  
[PIPPosition: PIPPosition] [OnMonitorRole: OnMonitorRole]

where

Mode: On/Off

Selfview is set to on or off.

FullscreenMode: On/Off

Choose between displaying the self-view in full screen or as picture-in-picture.

PIPPosition: CenterLeft/CenterRight/LowerLeft/LowerRight/UpperCenter/UpperLeft/UpperRight

Select the position for the self-view image.

OnMonitorRole: First/Fourth/Second/Third

Displays self-view on monitors with this role.

## Chapter 5

# xStatus commands

## Description of the xStatus commands

In this chapter, you can find all of the xStatus commands and the responses. Status type commands return information about the system and system processes. You can query all information or just some of it.

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

Go to: <http://www.cisco.com/go/dx-docs>

<b>Audio status</b> .....	<b>116</b>	<b>Call status</b> .....	<b>120</b>
xStatus Audio .....	116	xStatus Call [n].....	120
xStatus Audio Devices HandsetUSB Connection Status.....	116	xStatus Call [n] AnswerState .....	120
xStatus Audio Devices HandsetUSB Cradle .....	116	xStatus Call [n] AttendedTransferFrom.....	120
xStatus Audio Devices HeadsetAnalog Connection Status.....	116	xStatus Call [n] CallbackNumber .....	120
xStatus Audio Devices HeadsetAnalog HasMicrophone .....	116	xStatus Call [n] CallType .....	120
xStatus Audio Devices HeadsetUSB Connection Status.....	116	xStatus Call [n] DeviceType .....	120
xStatus Audio Devices HeadsetUSB Description .....	116	xStatus Call [n] Direction .....	120
xStatus Audio Devices HeadsetUSB Manufacturer .....	117	xStatus Call [n] DisplayName.....	121
xStatus Audio Input Connectors HDMI [n] EcReferenceDelay .....	117	xStatus Call [n] Duration .....	121
xStatus Audio Input Connectors Microphone [n] EcReferenceDelay .....	117	xStatus Call [n] Encryption Type .....	121
xStatus Audio Input RemotelInput [n] CallId .....	117	xStatus Call [n] FacilityServiceId .....	121
xStatus Audio Microphones Mute .....	117	xStatus Call [n] HoldReason .....	121
xStatus Audio Output Connectors Line [n] DelayMs .....	117	xStatus Call [n] PlacedOnHold.....	121
xStatus Audio SelectedDevice.....	118	xStatus Call [n] Protocol .....	122
xStatus Audio Volume .....	118	xStatus Call [n] ReceiveCallRate.....	122
xStatus Audio VolumeHandsetUsb .....	118	xStatus Call [n] RemoteNumber .....	122
xStatus Audio VolumeHeadsetAnalog .....	118	xStatus Call [n] Status.....	122
xStatus Audio VolumeHeadsetUsb .....	118	xStatus Call [n] TransmitCallRate.....	122
xStatus Audio VolumeInternal .....	118		
xStatus Audio VolumeMute .....	119	<b>Cameras status</b> .....	<b>123</b>
		xStatus Cameras Camera .....	123
<b>Bookings status</b> .....	<b>119</b>	xStatus Cameras Camera [n] Capabilities Options .....	123
xStatus Bookings Current .....	119	xStatus Cameras Camera [n] Connected.....	123
		xStatus Cameras Camera [n] MacAddress .....	123

xStatus Cameras Camera [n] Manufacturer .....	123	xStatus Conference Presentation LocalInstance [n] Source .....	129
xStatus Cameras Camera [n] Model .....	123	xStatus Conference Presentation Mode .....	129
xStatus Cameras Camera [n] SerialNumber .....	123	xStatus Conference SpeakerLock CallId .....	129
xStatus Cameras Camera [n] SoftwareID .....	124	xStatus Conference SpeakerLock Mode .....	129
<b>Capabilities status .....</b>	<b>124</b>	<b>Diagnostics status .....</b>	<b>130</b>
xStatus Capabilities .....	124	xStatus Diagnostics .....	130
xStatus Capabilities Conference MaxActiveCalls .....	124	xStatus Diagnostics Message [n] Description .....	130
xStatus Capabilities Conference MaxAudioCalls .....	124	xStatus Diagnostics Message [n] Level .....	130
xStatus Capabilities Conference MaxCalls .....	124	xStatus Diagnostics Message [n] References .....	130
xStatus Capabilities Conference MaxVideoCalls .....	125	xStatus Diagnostics Message [n] Type .....	130
<b>Conference status .....</b>	<b>125</b>	<b>H323 status .....</b>	<b>131</b>
xStatus Conference .....	125	xStatus H323 .....	131
xStatus Conference ActiveSpeaker CallId .....	125	xStatus H323 Gatekeeper Address .....	131
xStatus Conference Call [n] BookingId .....	125	xStatus H323 Gatekeeper Port .....	131
xStatus Conference Call [n] Capabilities FarendMessage Mode .....	125	xStatus H323 Gatekeeper Reason .....	131
xStatus Conference Call [n] Capabilities FECC Mode .....	126	xStatus H323 Gatekeeper Status .....	131
xStatus Conference Call [n] Capabilities FECC NumberOfPresets .....	126	xStatus H323 Mode Reason .....	131
xStatus Conference Call [n] Capabilities FECC NumberOfSources .....	126	xStatus H323 Mode Status .....	131
xStatus Conference Call [n] Capabilities FECC Source [n] Name .....	126	<b>HttpFeedback status .....</b>	<b>132</b>
xStatus Conference Call [n] Capabilities FECC Source [n] Options .....	126	xStatus HttpFeedback .....	132
xStatus Conference Call [n] Capabilities FECC Source [n] SourceId .....	126	xStatus HttpFeedback [1..4] Expression [1..15] .....	132
xStatus Conference Call [n] Capabilities Hold .....	127	xStatus HttpFeedback [1..4] Format .....	132
xStatus Conference Call [n] Capabilities IxChannel Status .....	127	xStatus HttpFeedback [1..4] URL .....	132
xStatus Conference Call [n] Capabilities Presentation .....	127	<b>MediaChannels status .....</b>	<b>132</b>
xStatus Conference Call [n] Manufacturer .....	127	xStatus MediaChannels .....	132
xStatus Conference Call [n] MicrophonesMuted .....	127	xStatus MediaChannels Call [n] Channel [n] Audio Channels .....	132
xStatus Conference Call [n] SoftwareID .....	127	xStatus MediaChannels Call [n] Channel [n] Audio Mute .....	132
xStatus Conference DoNotDisturb .....	128	xStatus MediaChannels Call [n] Channel [n] Audio Protocol .....	133
xStatus Conference Line [n] Mode .....	128	xStatus MediaChannels Call [n] Channel [n] Direction .....	133
xStatus Conference Multipoint Mode .....	128	xStatus MediaChannels Call [n] Channel [n] Encryption .....	133
xStatus Conference Presentation CallId .....	128	xStatus MediaChannels Call [n] Channel [n] NetStat Bytes .....	133
xStatus Conference Presentation LocalInstance [n] SendingMode .....	129		



- xStatus MediaChannels Call [n] Channel [n] NetStat ChannelRate ..... 133
- xStatus MediaChannels Call [n] Channel [n] NetStat Jitter..... 134
- xStatus MediaChannels Call [n] Channel [n] NetStat LastIntervalLost ..... 134
- xStatus MediaChannels Call [n] Channel [n] NetStat LastIntervalReceived ..... 134
- xStatus MediaChannels Call [n] Channel [n] NetStat Loss ..... 134
- xStatus MediaChannels Call [n] Channel [n] NetStat MaxJitter ..... 134
- xStatus MediaChannels Call [n] Channel [n] NetStat Packets ..... 134
- xStatus MediaChannels Call [n] Channel [n] ParticipantId ..... 135
- xStatus MediaChannels Call [n] Channel [n] Type ..... 135
- xStatus MediaChannels Call [n] Channel [n] Video ChannelRole ..... 135
- xStatus MediaChannels Call [n] Channel [n] Video FrameRate ..... 135
- xStatus MediaChannels Call [n] Channel [n] Video Protocol ..... 135
- xStatus MediaChannels Call [n] Channel [n] Video ResolutionX ..... 136
- xStatus MediaChannels Call [n] Channel [n] Video ResolutionY ..... 136
- xStatus MediaChannels Call [n] IncomingAudioChannel [n] Transport RTP Remote IpAddress ..... 136
- Network status ..... 136**
  - xStatus Network ..... 136
  - xStatus Network 1 CDP Address ..... 136
  - xStatus Network 1 CDP Capabilities ..... 136
  - xStatus Network 1 CDP Deviceld ..... 136
  - xStatus Network 1 CDP Duplex ..... 137
  - xStatus Network 1 CDP Platform ..... 137
  - xStatus Network 1 CDP PortID ..... 137
  - xStatus Network 1 CDP PrimaryMgmtAddress ..... 137
  - xStatus Network 1 CDP SysName ..... 137
  - xStatus Network 1 CDP SysObjectID ..... 137
  - xStatus Network 1 CDP Version ..... 138
  - xStatus Network 1 CDP VoIPApplianceVlanID ..... 138
  - xStatus Network 1 CDP VTPMgmtDomain ..... 138
  - xStatus Network 1 DNS Domain Name ..... 138
  - xStatus Network 1 DNS Server [1..5] Address ..... 138
  - xStatus Network 1 Ethernet MacAddress ..... 138
  - xStatus Network 1 Ethernet Speed ..... 139
  - xStatus Network 1 IPv4 Address ..... 139

- xStatus Network 1 IPv4 Gateway ..... 139
- xStatus Network 1 IPv4 SubnetMask ..... 139
- xStatus Network 1 IPv6 Address ..... 139
- xStatus Network 1 IPv6 Gateway ..... 139
- xStatus Network 1 VLAN Voice VlanId ..... 140
- NetworkServices status ..... 140**
  - xStatus NetworkServices ..... 140
  - xStatus NetworkServices NTP CurrentAddress ..... 140
  - xStatus NetworkServices NTP Server [n] Address ..... 140
  - xStatus NetworkServices NTP Status ..... 141
- Peripherals status ..... 141**
  - xStatus Peripherals ..... 141
  - xStatus Peripherals ConnectedDevice [n] HardwareInfo ..... 141
  - xStatus Peripherals ConnectedDevice [n] ID ..... 141
  - xStatus Peripherals ConnectedDevice [n] Name ..... 141
  - xStatus Peripherals ConnectedDevice [n] SoftwareInfo ..... 142
  - xStatus Peripherals ConnectedDevice [n] Status ..... 142
  - xStatus Peripherals ConnectedDevice [n] Type ..... 142
  - xStatus Peripherals ConnectedDevice [n] UpgradeStatus ..... 142
- Provisioning status ..... 143**
  - xStatus Provisioning ..... 143
  - xStatus Provisioning CUCM ExtensionMobility Enabled ..... 143
  - xStatus Provisioning CUCM ExtensionMobility LastLoggedInUserld ..... 143
  - xStatus Provisioning CUCM ExtensionMobility LoggedIn ..... 143
  - xStatus Provisioning Software Current CompletedAt ..... 143
  - xStatus Provisioning Software Current URL ..... 143
  - xStatus Provisioning Software Current VersionId ..... 143
  - xStatus Provisioning Software UpgradeStatus LastChange ..... 144
  - xStatus Provisioning Software UpgradeStatus Message ..... 144
  - xStatus Provisioning Software UpgradeStatus Phase ..... 144
  - xStatus Provisioning Software UpgradeStatus SessionId ..... 144
  - xStatus Provisioning Software UpgradeStatus Status ..... 144



- xStatus Provisioning Software UpgradeStatus URL ..... 144
- xStatus Provisioning Software UpgradeStatus VersionId ..... 145
- xStatus Provisioning Status..... 145
  
- Proximity status..... 145**
  - xStatus Proximity..... 145
  - xStatus Proximity Services Availability..... 145
  
- Security status..... 146**
  - xStatus Security ..... 146
  - xStatus Security FIPS Mode..... 146
  - xStatus Security Persistency CallHistory ..... 146
  - xStatus Security Persistency Configurations..... 146
  - xStatus Security Persistency DHCP..... 146
  - xStatus Security Persistency InternalLogging..... 146
  - xStatus Security Persistency LocalPhonebook ..... 146
  
- SIP status ..... 147**
  - xStatus SIP ..... 147
  - xStatus SIP AlternateURI Alias [n] URI ..... 147
  - xStatus SIP AlternateURI Primary [n] URI ..... 147
  - xStatus SIP Authentication ..... 147
  - xStatus SIP CallForward DisplayName..... 147
  - xStatus SIP CallForward Mode ..... 147
  - xStatus SIP CallForward URI ..... 147
  - xStatus SIP Mailbox MessagesWaiting ..... 148
  - xStatus SIP Mailbox URI ..... 148
  - xStatus SIP Proxy [n] Address..... 148
  - xStatus SIP Proxy [n] Status ..... 148
  - xStatus SIP Registration [n] Authentication ..... 148
  - xStatus SIP Registration [n] Reason ..... 149
  - xStatus SIP Registration [n] Status ..... 149
  - xStatus SIP Registration [n] URI..... 149
  - xStatus SIP Secure..... 149
  - xStatus SIP Verified..... 149

- Standby status..... 150**
  - xStatus Standby ..... 150
  - xStatus Standby State..... 150
  
- SystemUnit status..... 150**
  - xStatus SystemUnit ..... 150
  - xStatus SystemUnit Hardware Module CompatibilityLevel..... 150
  - xStatus SystemUnit Hardware Module SerialNumber ..... 150
  - xStatus SystemUnit Hardware Monitoring Fan [n] Status..... 151
  - xStatus SystemUnit Hardware Temperature..... 151
  - xStatus SystemUnit Notifications Notification [n] Text ..... 151
  - xStatus SystemUnit Notifications Notification [n] Type ..... 151
  - xStatus SystemUnit ProductId ..... 151
  - xStatus SystemUnit ProductPlatform..... 152
  - xStatus SystemUnit ProductType ..... 152
  - xStatus SystemUnit Software DisplayName ..... 152
  - xStatus SystemUnit Software Name..... 152
  - xStatus SystemUnit Software OptionKeys RemoteMonitoring ..... 152
  - xStatus SystemUnit Software ReleaseDate ..... 152
  - xStatus SystemUnit Software Version ..... 153
  - xStatus SystemUnit State NumberOfActiveCalls..... 153
  - xStatus SystemUnit State NumberOfInProgressCalls ..... 153
  - xStatus SystemUnit State NumberOfSuspendedCalls ..... 153
  - xStatus SystemUnit Uptime ..... 153
  
- Time status ..... 154**
  - xStatus Time..... 154
  - xStatus Time SystemTime..... 154
  
- UserInterface status ..... 154**
  - xStatus UserInterface..... 154
  - xStatus UserInterface ContactInfo ContactMethod [n] Number ..... 154
  - xStatus UserInterface ContactInfo Name ..... 154
  - xStatus UserInterface Extensions Widget [n] WidgetId ..... 154
  - xStatus UserInterface Extensions Widget [n] Value ..... 155

<b>Video status.....</b>	<b>155</b>	<b>xStatus Video Monitors .....</b>	<b>158</b>
xStatus Video .....	155	xStatus Video Output .....	158
xStatus Video ActiveSpeaker PIPPosition.....	155	xStatus Video Output Connector [n] Connected.....	158
xStatus Video Input .....	155	xStatus Video Output Connector [n] ConnectedDevice CEC DeviceType.....	158
xStatus Video Input Connector [n] Connected.....	155	xStatus Video Output Connector [n] ConnectedDevice CEC PowerControl .....	159
xStatus Video Input Connector [n] SignalState .....	156	xStatus Video Output Connector [n] ConnectedDevice CEC PowerStatus.....	159
xStatus Video Input Connector [n] SourceId.....	156	xStatus Video Output Connector [n] ConnectedDevice Name .....	159
xStatus Video Input Connector [n] Type.....	156	xStatus Video Output Connector [n] ConnectedDevice PreferredFormat.....	159
xStatus Video Input MainVideoSource.....	156	xStatus Video Output Connector [n] MonitorRole .....	159
xStatus Video Input Source [n] ConnectorId .....	156	xStatus Video Output Connector [n] Resolution Height .....	159
xStatus Video Input Source [n] FormatStatus .....	156	xStatus Video Output Connector [n] Resolution RefreshRate.....	160
xStatus Video Input Source [n] FormatType .....	157	xStatus Video Output Connector [n] Resolution Width .....	160
xStatus Video Input Source [n] MediaChannelId.....	157	xStatus Video Output Connector [n] Type.....	160
xStatus Video Input Source [n] Resolution Height.....	157	xStatus Video Presentation PIPPosition.....	160
xStatus Video Input Source [n] Resolution RefreshRate .....	157	xStatus Video Selfview FullscreenMode.....	160
xStatus Video Input Source [n] Resolution Width.....	157	xStatus Video Selfview Mode .....	160
xStatus Video Layout LayoutFamily Local.....	157	xStatus Video Selfview OnMonitorRole .....	161
xStatus Video Layout LayoutFamily Remote .....	158	xStatus Video Selfview PIPPosition .....	161

## 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 Devices HandsetUSB Connection Status

Shows if a USB handset is connected.

**Value space of the result returned:**

Connected/NotConnected

**Example:**

```
xStatus Audio Devices HandsetUSB Connection Status
*s Connected
** end
```

### xStatus Audio Devices HandsetUSB Cradle

Shows if a USB handset is on hook or off hook.

**Value space of the result returned:**

OnHook/OffHook

**Example:**

```
xStatus Audio Devices HandsetUSB Cradle
*s OnHook
** end
```

### xStatus Audio Devices HeadsetAnalog Connection Status

Shows if an analog headset is connected. (DX70 only)

**Value space of the result returned:**

Connected/NotConnected

**Example:**

```
xStatus Audio Devices HeadsetAnalog Connection Status
*s Connected
** end
```

### xStatus Audio Devices HeadsetAnalog HasMicrophone

Shows if an analog headset has a built-in microphone. (DX70 only)

**Value space of the result returned:**

False/True

**Example:**

```
xStatus Audio Devices HeadsetAnalog HasMicrophone
*s True
** end
```

### xStatus Audio Devices HeadsetUSB Connection Status

Shows if a USB headset is connected.

**Value space of the result returned:**

Connected/NotConnected

**Example:**

```
xStatus Audio Devices HeadsetUSB Connection Status
*s Connected
** end
```

### xStatus Audio Devices HeadsetUSB Description

Shows a description of the attached USB headset.

**Value space of the result returned:**

String

**Example:**

```
xStatus Audio Devices HeadsetUSB Description
*s HeadsetUSB Description: "Description of USB headset."
** end
```

### xStatus Audio Devices HeadsetUSB Manufacturer

Lists the USB headset manufacturer.

**Value space of the result returned:**

String

**Example:**

```
xStatus Audio Devices HeadsetUSB Manufacturer
*s HeadsetUSB Manufacturer: "CompanyName"
** end
```

### xStatus Audio Input Connectors HDMI [n] EcReferenceDelay

Returns the detected latency for each loudspeaker to microphone path for systems supporting HDMI input used as microphone input. The result is in milliseconds.

**Value space of the result returned:**

Integer

**Example:**

```
xStatus Audio Input Connectors HDMI 1 EcReferenceDelay
*s Audio Connectors HDMI 1 EcReferenceDelay: 0
** 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 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 Microphones Mute

Shows whether the microphones are muted.

**Value space of the result returned:**

On/Off

**Example:**

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

### xStatus Audio Output Connectors Line [n] DelayMs

Shows the delay in milliseconds.

**Value space of the result returned:**

Integer (0..290)

**Example:**

```
xStatus Audio Output Connectors Line 1
*s Audio Output Connectors Line 1 DelayMs: 60
** end
```

### xStatus Audio SelectedDevice

Shows the audio device that is selected. Only the DX70 supports analog headsets.

**Value space of the result returned:**

Internal/HeadsetUSB/HeadsetAnalog/HandsetUSB

**Example:**

```
xStatus Audio SelectedDevice
*s HeadsetUSB
** 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 VolumeHandsetUsb

Shows the volume level of the USB handset.

**Value space of the result returned:**

Integer

**Example:**

```
xStatus Audio VolumeHandsetUsb
*s 6
** end
```

### xStatus Audio VolumeHeadsetAnalog

Shows the volume level of the analog headset (DX70 only).

**Value space of the result returned:**

Integer

**Example:**

```
xStatus Audio VolumeHeadsetAnalog
*s 6
** end
```

### xStatus Audio VolumeHeadsetUsb

Shows the volume level of the USB headset.

**Value space of the result returned:**

Integer

**Example:**

```
xStatus Audio VolumeHeadsetUsb
*s 6
** end
```

### xStatus Audio VolumeInternal

Shows the internal volume level.

**Value space of the result returned:**

Integer

**Example:**

```
xStatus Audio VolumeInternal
*s 6
** 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
```

## Bookings status

### xStatus Bookings Current

The ID of the on going booking event, if any.

**Value space of the result returned:**

String

**Example:**

```
xStatus Bookings Current Id
*s Bookings Current Id: "123"
** end
```

## Call status

### xStatus Call [n]

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] 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] 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
```

### xStatus Call [n] CallbackNumber

Shows the remote (far end) number or URI of an incoming or outgoing call, including the call protocol, for call back. You can run the command xStatus Call to find the call identity.

**Value space of the result returned:**

String

**Example:**

```
xStatus Call 27 CallbackNumber
*s Call 27 CallbackNumber: "h323:firstname.lastname@company.com"
** end
```

### xStatus Call [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] DeviceType

Indicates whether the remote system is a single endpoint or an MCU. Some Cisco endpoints (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] 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] 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] 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] 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] 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] HoldReason

Shows the reason the current outgoing call was put on hold.

Conference: On hold while the call is being merged into a conference.

Transfer: On hold while the call is being transferred.

None: All other instances.

**Value space of the result returned:**

Conference/Transfer/None

**Example:**

```
xStatus Call 2 HoldReason
*s Call 2 HoldReason: 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] 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:**

H320/H323/SIP/Spark/Unknown

**Example:**

```
xStatus Call 27 Protocol
*s Call 27 Protocol: "h323"
** 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] 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] 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] 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
```

## Cameras status

### xStatus Cameras Camera

Shows the top level overview of the camera status.

### xStatus Cameras Camera [n] Capabilities Options

Shows the camera capabilities (ptzf = pan, tilt, zoom, focus).

**Value space of the result returned:**

String

**Example:**

```
xStatus Cameras Camera 1 Capabilities Options
*s Camera 1 Capabilities Options: "ptzf"
** end
```

### xStatus Cameras Camera [n] Connected

Shows if the camera is connected or not.

**Value space of the result returned:**

True/False

**Example:**

```
xStatus Cameras Camera 1 Connected
*s Camera 1 Connected: True
** end
```

### xStatus Cameras Camera [n] MacAddress

Shows the MAC (Media Access Control) address for the camera.

**Value space of the result returned:**

String

**Example:**

```
xStatus Cameras Camera 1 MacAddress
*s Camera 1 MacAddress: ""
** end
```

### xStatus Cameras Camera [n] Manufacturer

Shows the manufacturer of the camera.

**Value space of the result returned:**

String

**Example:**

```
xStatus Cameras Camera 1 Manufacturer
*s Camera 1 Manufacturer: "Cisco"
** end
```

### xStatus Cameras Camera [n] Model

Shows the camera model.

**Value space of the result returned:**

String

**Example:**

```
xStatus Cameras Camera 1 Model
*s Camera 1 Model: "Precision 40"
** end
```

### xStatus Cameras Camera [n] SerialNumber

Shows the camera serial number.

**Value space of the result returned:**

String

**Example:**

```
xStatus Cameras Camera 1 SerialNumber
*s Camera 1 SerialNumber: "B1AB26B00010"
** end
```

### xStatus Cameras Camera [n] SoftwareID

Shows the software identity of the camera.

**Value space of the result returned:**

String

**Example:**

```
xStatus Cameras Camera 1 SoftwareID
*s Camera 1 SoftwareID: "S01718-4.0FINAL [ID:40063] 2014-10-20"
** end
```

## Capabilities status

### xStatus Capabilities

Shows the top level overview of the capabilities status.

### xStatus Capabilities Conference MaxActiveCalls

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 Capabilities Conference MaxNumberOfActiveCalls
*s Capabilities Conference MaxNumberOfActiveCalls: 3
** end
```

### xStatus Capabilities Conference MaxAudioCalls

Shows the the maximum number of simultaneous audio calls that is supported.

**Value space of the result returned:**

Integer

**Example:**

```
xStatus Capabilities Conference MaxAudioCalls
*s Capabilities Conference MaxAudioCalls: 3
** end
```

### xStatus Capabilities Conference MaxCalls

Shows the the maximum number of simultaneous calls.

**Value space of the result returned:**

0..5

**Example:**

```
xStatus Capabilities Conference MaxCalls
*s Capabilities Conference MaxCalls: 3
** end
```

### xStatus Capabilities Conference MaxVideoCalls

Shows the the maximum number of simultaneous video calls that is supported.

**Value space of the result returned:**

Integer

**Example:**

```
xStatus Capabilities Conference MaxVideoCalls
*s Capabilities Conference MaxVideoCalls: 3
** end
```

## Conference status

### xStatus Conference

Shows the top level overview of the conference status. The identity of the Conference Call can only be read during a call.

### xStatus Conference ActiveSpeaker CallId

Shows the CallId of the current active speaker.

**Value space of the result returned:**

Integer

**Example:**

```
xStatus Conference ActiveSpeaker CallId
*s Conference ActiveSpeaker CallId: 3
** end
```

### xStatus Conference Call [n] BookingId

Shows the booking ID of a conference (if assigned). The booking ID can be used for easy identification of a call or conference.

**Value space of the result returned:**

String

**Example:**

```
xStatus Conference Call 2 BookingId
*s Conference Call 2 BookingId: "MyConference"
** end
```

### xStatus Conference Call [n] Capabilities FarendMessage Mode

Not applicable in this release.

**Value space of the result returned:**

On/Off

**Example:**

```
xStatus Conference Call Capabilities FarendMessage Mode
*s Conference Call 4 Capabilities FarendMessage Mode: Off
** end
```

### xStatus Conference Call [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 Call 2 Capabilities FECC Mode
*s Conference Call 2 Capabilities FECC Mode: On
** end
```

### xStatus Conference Call [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 Call 2 Capabilities FECC NumberOfPresets
*s Conference Call 2 Capabilities FECC NumberOfPresets: 15
** end
```

### xStatus Conference Call [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 Call 2 Capabilities FECC NumberOfSources
*s Conference Call 2 Capabilities FECC NumberOfSources: 5
** end
```

### xStatus Conference Call [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 Call 2 Capabilities FECC Source 1 Name
*s Conference Call 2 Capabilities FECC Source 1 Name: "Main camera"
** end
```

### xStatus Conference Call [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 Call 2 Capabilities FECC Source 1 Options
*s Conference Call 2 Capabilities FECC Source 1 Options: "ptzf"
** end
```

### xStatus Conference Call [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 Call 2 Capabilities FECC Source 1 SourceId
*s Conference Call 2 Capabilities FECC Source 1 SourceId: 6
** end
```

### xStatus Conference Call [n] Capabilities 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 Call Capabilities Hold
*s Conference Call 2 Capabilities Hold: True
** end
```

### xStatus Conference Call [n] Capabilities IxChannel Status

Not applicable in this release.

**Value space of the result returned:**

Active/Failed/Off

**Example:**

```
xStatus Conference Call 4 Capabilities IxChannel Status
*s Conference Call 4 Capabilities IxChannel Status: Active
** end
```

### xStatus Conference Call [n] Capabilities Presentation

Lists the presentation capabilities for other participants in the conference.

**Value space of the result returned:**

True/False

**Example:**

```
xStatus Conference Call 2 Capabilities Presentation
*s Conference Call 2 Capabilities Presentation: True
** end
```

### xStatus Conference Call [n] Manufacturer

Shows the manufacturer of the video system at a far end site.

**Value space of the result returned:**

String

**Example:**

```
xStatus Conference Call 2 Manufacturer
*s Conference Call 2 Manufacturer: "Cisco"
** end
```

### xStatus Conference Call [n] MicrophonesMuted

Lists the audio mute status for other participants in the conference.

**Value space of the result returned:**

True/False

**Example:**

```
xStatus Conference Call 2 MicrophonesMuted
*s Conference Call 2 MicrophonesMuted: True
** end
```

### xStatus Conference Call [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 Call 2 SoftwareID
*s Conference Call 2 SoftwareID: "CE8"
** 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 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 Multipoint Mode

Shows how the Multipoint video conferences are handled.

Auto: The multipoint method available will be chosen 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.

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.

**Value space of the result returned:**

Auto/CUCMMediaResourceGroupList/MultiSite/Off

**Example:**

```
xStatus Conference Multipoint Mode
*s Conference Multipoint Mode: "Auto"
** end
```

### xStatus Conference Presentation CallId

Shows the identity of the system that sends the presentation.

**Value space of the result returned:**

Integer

**Example:**

```
xStatus Conference Presentation CallId
*s Conference Presentation CallId: 0
** end
```



### xStatus Conference Presentation LocalInstance [n] SendingMode

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 LocalInstance 1 SendingMode
*s Conference Presentation LocalInstance 1 SendingMode: LocalOnly
** end
```

### xStatus Conference Presentation LocalInstance [n] Source

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 LocalInstance 1 Source
*s Conference Presentation LocalInstance 1 Source: 1
** end
```

### xStatus Conference Presentation Mode

Shows the status of the secondary video stream.

**Value space of the result returned:**

On/Off

**Example:**

```
xStatus Conference Presentation Mode
*s Conference Presentation Mode: Off
** end
```

### xStatus Conference SpeakerLock CallId

Shows the CallId for the participant locked as the prominent speaker in the conference.

**Value space of the result returned:**

Integer

**Example:**

```
xStatus Conference SpeakerLock CallId
*s Conference SpeakerLock CallId: 0
** end
```

### xStatus Conference SpeakerLock Mode

Shows whether a speaker lock is set or not.

**Value space of the result returned:**

On/Off

**Example:**

```
xStatus Conference SpeakerLock Mode
*s Conference SpeakerLock Mode: Off
** 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 Message [n] Description

Shows a description of the current diagnostics alerts.

#### Value space of the result returned:

String

#### Example:

```
xStatus Diagnostics Message Description
*s DiagnosticsResult Message 1 Description: "IP configuration incomplete"
** end
```

### xStatus Diagnostics Message [n] Level

Shows the level of importance 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 Diagnostics Message 4 Level
*s Diagnostics Message 4 Level: Warning
** end
```

### xStatus Diagnostics Message [n] References

Additional information on the diagnostics alert, if available.

#### Value space of the result returned:

String

#### Example:

```
xStatus Diagnostics Message 10 References
*s Diagnostics Message 10 References: "delay=190"
** end
```

### xStatus Diagnostics Message [n] Type

Shows information on the results of the latest diagnostics on the system.

#### Value space of the result returned:

CAPFOperationState/CTLInstallation/CUCMVendorConfigurationFile/CallProtocolDualStackConfig/CallProtocolIPStackPlatformCompatibility/CallProtocolVcsProvisioningCompatibility/CameraId/CameraPairing/CameraSerial/CameraSoftwareVersion/CameraStatus/CamerasDetected/ConfigurationFile/DefaultCallProtocolRegistered/EthernetDuplexMatches/FanStatus/PresenterTrack/H320GatewayStatus/H323GatekeeperStatus/HasValidReleaseKey/IpCameraStatus/IPv4Assignment/IPv6Assignment/IPv6Mtu/ISDNLinkCompatibility/ISDNLinkIpStack/ITLInstallation/InvalidSIPTransportConfig/LockDown/RoomControl/MediaBlockingDetected/NetLinkStatus/NetSpeedAutoNegotiated/NetworkQuality/NTPStatus/OSDVideoOutput/OutputConnectorLocations/ProvisioningStatus/SIPEncryption/SIPListenPortAndOutboundMode/SIPProfileRegistration/SIPProfileType/SelectedVideoInputSourceConnected/SipIceAndAnatConflict/TCPMediaFallback/TemperatureCheck/TLSVerifyRequiredCerts/TouchPanelConnection/ControlSystemConnection/TurnBandwidth/UdpPortRangeViolation/UltrasoundSpeakerAvailability/ValidPasswords/VideoFromInternalCamera/VideoInputStability/SparkConnectivity/SpeakerTrackFrontPanelMountedCorrectly/SpeakerTrackMicrophoneConnection/SpeakerTrackVideoInputs/SpeakerTrackEthernetConnection/ANATonVCS/ECReferenceDelay/AudioECRefDelay/AudioPairingNoise/AudioPairingRate/AudioPairingSNR/AudioPairingInterference/AudioPairingTokenDecode/AudioInternalSpeakerDisabled/AbnormalCallTermination/HasActiveCallProtocol/SipOrH323ButNotBothEnabled/PresentationSourceSelection/SoftwareUpgrade/PlatformSanity/ContactInfoMismatch/MicrophoneReinforcement

#### Example:

```
xStatus Diagnostics Message Type
*s Diagnostics Message 1 Type: CamerasDetected
** end
```

## H323 status

### xStatus H323

Shows the top level overview of the H323 status.

### 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 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 Mode Reason

Shows whether there is a conflict between H.323 settings and xStatus H323 Mode Status.

“”: When H.323 is set to On and there is no conflict between H.323 Mode configuration and the rest of the system settings.

“SIP is enabled”: When H.323 Mode is set to On and SIP is enabled on a system that does not support the two simultaneously.

“Not available”: When a system does not support H.323.

**Value space of the result returned:**

String

**Example:**

```
xStatus H323 Mode Reason
*s H323 Mode Reason: ""
** end
```

### xStatus H323 Mode Status

Shows the status for H.323 registration.

Enabled: Registration is enabled.

Disabled: Registration is disable, because SIP is enabled.

**Value space of the result returned:**

Enabled/Disabled

**Example:**

```
xStatus H323 Mode Status
*s H323 Mode Status: "Disabled"
** end
```

## HttpFeedback status

### xStatus HttpFeedback

Shows the top level overview of the HTTP status.

### 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 HttpFeedback [1..4] Format

Shows the format of the HTTP Feedback feedback from the HTTP server.

#### Value space of the result returned:

XML/JSON

#### Example:

```
xStatus HttpFeedback 1 Format
*s HttpFeedback 1 Format: XML
** end
```

### 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
```

## 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] Channel [n] Audio Channels

Shows the number of incoming or outgoing audio channels.

#### Value space of the result returned:

Integer

#### Example:

```
xStatus MediaChannels Call 27 Channel 327 Audio Channels
*s MediaChannels Call 27 Channel 327 Audio Channels: 1
** end
```

### xStatus MediaChannels Call [n] Channel [n] Audio Mute

Audio mute status of the incoming audio.

#### Value space of the result returned:

True/False

#### Example:

```
xStatus MediaChannels Call 27 Channel 327 Audio Mute
*s MediaChannels Call 27 Channel 327 Audio Mute: False
** end
```

### xStatus MediaChannels Call [n] Channel [n] Audio Protocol

Shows the audio algorithm of the incoming or outgoing audio.

AACLD: The AAC-LD is an MPEG-4 Low Delay Audio Coder audio compression format.

G711A: The G.711 A-law algorithm is an ITU-T standard for audio compression.

G711Mu: The G.711 Mu-law algorithm is an ITU-T standard for audio compression.

G722: The G.722 algorithm is an ITU-T standard for audio compression.

G7221: The G.722.1 algorithm is an ITU-T standard for audio compression.

G7221C: The G.722.1 annex C algorithm is an ITU-T standard for audio compression.

G723\_1: The G.723.1 algorithm is an ITU-T standard for audio compression.

G728: The G.728 algorithm is an ITU-T standard for audio compression.

G729: The G.729 algorithm is an ITU-T standard for audio compression.

G729A: The G.729 annex A algorithm is an ITU-T standard for audio compression.

G729AB: The G.729 annex A and B algorithm is an ITU-T standard for audio compression.

Off: No audio.

Opus: Opus is a royalty-free IETF standard for audio compression.

**Value space of the result returned:**

AACLD/G711A/G711Mu/G722/G7221/G7221C/G723\_1/G728/G729/G729A/G729AB/Off/Opus

**Example:**

```
xStatus MediaChannels Call 27 Channel 327 Audio Protocol
*s MediaChannels Call 27 Channel 327 Audio Protocol: AACLD
** end
```

### xStatus MediaChannels Call [n] Channel [n] Direction

Shows the direction of the call.

Incoming: The call is an incoming call.

Outgoing: The call is an outgoing call.

**Value space of the result returned:**

Incoming/Outgoing

**Example:**

```
xStatus MediaChannels Call 27 Channel 327 Direction
*s MediaChannels Call 27 Channel 327 Direction: Incoming
** end
```

### xStatus MediaChannels Call [n] Channel [n] Encryption

Shows the encryption status for audio or video on the incoming or outgoing call.

**Value space of the result returned:**

On/Off

**Example:**

```
xStatus MediaChannels Call 27 Channel 327 Encryption
*s MediaChannels Call 27 Channel 327 Encryption: On
** end
```

### xStatus MediaChannels Call [n] Channel [n] NetStat Bytes

Shows the number of bytes for audio, video or data on the incoming or outgoing channel.

**Value space of the result returned:**

Integer

**Example:**

```
xStatus MediaChannels Call 27 Channel 327 NetStat Bytes
*s MediaChannels Call 27 Channel 327 NetStat Bytes: 496640
** end
```

### xStatus MediaChannels Call [n] Channel [n] NetStat ChannelRate

Shows the bandwidth for audio, video or data on the incoming or outgoing channel.

**Value space of the result returned:**

Integer

**Example:**

```
xStatus MediaChannels Call 27 Channel 327 NetStat ChannelRate
*s MediaChannels Call 27 Channel 327 NetStat ChannelRate: 128000
** end
```

### xStatus MediaChannels Call [n] Channel [n] NetStat Jitter

Shows the jitter for audio, video or data at the present moment on the incoming or outgoing channel, as specified by RFC 3550.

**Value space of the result returned:**

Integer

**Example:**

```
xStatus MediaChannels Call 27 Channel 327 NetStat Jitter
*s MediaChannels Call 27 Channel 327 NetStat Jitter: 2
** end
```

### xStatus MediaChannels Call [n] Channel [n] NetStat LastIntervalLost

Shows the number of packets lost for audio, video or data during the last interval on the incoming or outgoing channels.

**Value space of the result returned:**

Integer

**Example:**

```
xStatus MediaChannels Call 27 Channel 327 NetStat LastIntervalLost
*s MediaChannels Call 27 Channel 327 NetStat LastIntervalLost: 0
** end
```

### xStatus MediaChannels Call [n] Channel [n] NetStat LastIntervalReceived

Shows the number of packets received for audio, video or data during the last interval on the incoming or outgoing channels.

**Value space of the result returned:**

Integer

**Example:**

```
xStatus MediaChannels Call 27 Channel 327 NetStat LastIntervalReceived
*s MediaChannels Call 27 Channel 327 NetStat LastIntervalReceived: 289
** end
```

### xStatus MediaChannels Call [n] Channel [n] NetStat Loss

Shows the number of packets lost for audio, video or data on the incoming or outgoing channels.

**Value space of the result returned:**

Integer

**Example:**

```
xStatus MediaChannels Call 27 Channel 327 NetStat Loss
*s MediaChannels Call 27 Channel 327 NetStat Loss: 0
** end
```

### xStatus MediaChannels Call [n] Channel [n] NetStat MaxJitter

Shows the maximum jitter for audio, video or data that has been measured during last interval (about 5 seconds).

**Value space of the result returned:**

Integer

**Example:**

```
xStatus MediaChannels Call 27 Channel 327 NetStat MaxJitter
*s MediaChannels Call 27 Channel 327 NetStat MaxJitter: 2
** end
```

### xStatus MediaChannels Call [n] Channel [n] NetStat Packets

Shows the number of packets that was received or sent for audio, video or data on the incoming or outgoing channels.

**Value space of the result returned:**

Integer

**Example:**

```
xStatus MediaChannels Call 27 Channel 327 NetStat Packets
*s MediaChannels Call 27 Channel 327 NetStat Packets: 1551
** end
```

### xStatus MediaChannels Call [n] Channel [n] ParticipantId

Shows the ID of the Active Control participant on the incoming audio or video channel.

**Value space of the result returned:**

String

**Example:**

```
xStatus MediaChannels Call 27 Channel 327 ParticipantId
*s MediaChannels Call 27 Channel 327 ParticipantId: ""
** end
```

### xStatus MediaChannels Call [n] Channel [n] Type

Shows the media type on the incoming or outgoing channel.

Audio: The media type on the incoming or outgoing channel is audio.

Video: The media type on the incoming or outgoing channel is video.

Data: The media type on the incoming or outgoing channel is data.

**Value space of the result returned:**

Audio/Video/Data

**Example:**

```
xStatus MediaChannels Call 27 Channel 327 Type
*s MediaChannels Call 27 Channel 327 Type: Audio
** end
```

### xStatus MediaChannels Call [n] Channel [n] Video ChannelRole

Shows the role of the incoming or outgoing video channel.

Main: The main video is on the incoming or outgoing video channel.

Presentation: The presentation is on the incoming or outgoing video channel.

**Value space of the result returned:**

Main/Presentation

**Example:**

```
xStatus MediaChannels Call 27 Channel 327 Video ChannelRole
*s MediaChannels Call 27 Channel 327 Video ChannelRole: Main
** end
```

### xStatus MediaChannels Call [n] Channel [n] Video FrameRate

Shows the frame rate of the incoming or outgoing video channel.

**Value space of the result returned:**

Integer

**Example:**

```
xStatus MediaChannels Call 27 Channel 327 Video FrameRate
*s MediaChannels Call 27 Channel 327 Video FrameRate: 30
** end
```

### xStatus MediaChannels Call [n] Channel [n] Video Protocol

Shows the video algorithm of the incoming or outgoing video channel.

Shows the video algorithm on the incoming/outgoing video channel.

H261: The H.261 algorithm is an ITU-T standard for video compression.

H263: The H.263 algorithm is an ITU-T standard for video compression.

H263p: The H.263p (H.263+) algorithm is an ITU-T standard for video compression.

H263pp: The H.263pp (H.263++) algorithm is an ITU-T standard for video compression.

H264: The H.264 algorithm (MPEG-4 AVC) is an ITU-T and ISE/IEC standard for video compression.

HEVC: The HEVC (H.265) algorithm (MPEG-H part 2) is an ITU-T and ISE/IEC standard for video compression.

JPEG: The JPEG algorithm is an ISO/IEC standard for video compression.

MP4V: The MP4V format is an ISO/IEC file format for video compression.

Off: No video.

**Value space of the result returned:**

H261/H263/H263p/H263pp/H264/HEVC/JPEG/MP4V/Off

**Example:**

```
xStatus MediaChannels Call 27 Channel 327 Video Protocol
*s MediaChannels Call 27 Channel 327 Video Protocol: H264
** end
```

### xStatus MediaChannels Call [n] Channel [n] Video ResolutionX

Shows the width (resolution in direction X) of the incoming or outgoing video.

**Value space of the result returned:**

Integer

**Example:**

```
xStatus MediaChannels Call 27 Channel 327 Video ResolutionX
*s MediaChannels Call 27 Channel 327 Video ResolutionX: 1920
** end
```

### xStatus MediaChannels Call [n] Channel [n] Video ResolutionY

Shows the width (resolution in direction Y) of the incoming or outgoing video.

**Value space of the result returned:**

Integer

**Example:**

```
xStatus MediaChannels Call 27 Channel 327 Video ResolutionY
*s MediaChannels Call 27 Channel 327 Video ResolutionY: 1080
** 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
```

## Network status

### xStatus Network

Shows the top level overview of the network status.

### 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 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 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 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 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 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 Version

Returns information about the software release version the switch is running.

**Value space of the result returned:**

String

**Example:**

```
xStatus Network 1 CDP Version
*s Network 1 CDP Version: "Cisco IOS Software, C3560CX Software (C3560CX-
UNIVERSALK9-M), Version 15.2(3)E, RELEASE SOFTWARE (fc4)*Technical Support:
http://www.cisco.com/techsupport*Copyright (c) 1986-2014 by Cisco Systems,
Inc.*Compiled Sun 07-Dec-14 13:15 by prod_rel_team"
** 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 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 DNS Domain Name

Shows the domain name.

**Value space of the result returned:**

String

**Example:**

```
xStatus Network 1 DNS Domain Name
*s Network 1 DNS Domain Name: "www.example.com www.example.int"
** end
```

### xStatus Network 1 DNS Server [1..5] Address

Shows the IP address of the DNS server.

**Value space of the result returned:**

String

**Example:**

```
xStatus Network 1 DNS Server 1. Address
*s Network 1 DNS Server 1 Address: "192.0.2.60"
** end
```

### 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 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 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 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
```

## NetworkServices status

### xStatus NetworkServices

Shows the top level overview of the network services status.

### 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 Server [n] 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 Status

Returns the status of the endpoints synchronizing with the NTP server.

Discarded: The NTP result has been discarded.

Synced: The system is in sync with the NTP server.

NotSynced: The system is not in sync with the NTP server.

Unknown: The state of the synchronization is unknown.

Off: No synchronization with the NTP server.

**Value space of the result returned:**

Discarded/Synced/NotSynced/Unknown/Off

**Example:**

```
xStatus NetworkServices NTP Status
*s NetworkServices NTP Status: Synced
** end
```

## Peripherals status

### xStatus Peripherals

Shows the top level overview of the 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.

**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:**

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/InstallationReady/None/Succeeded/Rebooting/Retrying/Aborted/Paused

**Example:**

```
xStatus Peripherals ConnectedDevice 1001 UpgradeStatus
*s Peripherals ConnectedDevice 1001 UpgradeStatus: None
** end
```

## Provisioning status

### xStatus Provisioning

Shows the top level overview of the provisioning status.

### xStatus Provisioning CUCM ExtensionMobility Enabled

Shows if Extension Mobility is enabled for a CUCM 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 LastLoggedInUserId

Returns the user id that was last logged in to the system. Only in use if CUCM Extension Mobility is enabled, and CUCM 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 ExtensionMobility LoggedIn

Shows if someone is 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 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 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://.../s52020ce8_0_0.pkg"
** 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: "s52020ce8_0_0.pkg"
** 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 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 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
```

### 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 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 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://.../s52020ce8_0_0.pkg"
** 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: "s52010ce8_0_0.pkg"
** end
```

### 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
```

## Proximity status

### xStatus Proximity

Shows the top level overview of the proximity status.

### xStatus Proximity Services Availability

Shows whether proximity services are available on the endpoint.

Available: Proximity mode has been enabled with the command xConfiguration Proximity Mode and one or more of the proximity services have been enabled with xConfiguration Proximity Services commands.

Disabled: Proximity mode has been disabled with xConfiguration Proximity Mode, or none of the services have been enabled with the xConfiguration Proximity Services commands.

Deactivated: Proximity services have been deactivated with the command xCommand Proximity Services Deactivate.

**Example:**

```
xStatus Proximity Services Availability
*s Proximity Services Availability: Disabled
** 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 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 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 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 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
```

## SIP status

### xStatus SIP

Shows the top level overview of the SIP status.

### xStatus SIP AlternateURI Alias [n] URI

**Value space of the result returned:**

String

**Example:**

```
xStatus SIP AlternateURI Alias
*s SIP AlternateURI Alias URI: ""
** end
```

### xStatus SIP AlternateURI Primary [n] URI

**Value space of the result returned:**

String

**Example:**

```
xStatus SIP AlternateURI Primary
*s SIP AlternateURI Primary URI: ""
** end
```

### xStatus SIP 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 Authentication
*s SIP Authentication: Off
** end
```

### xStatus SIP 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 CallForward DisplayName
*s SIP CallForward DisplayName: ""
** end
```

### xStatus SIP 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 CallForward Mode
*s SIP CallForward Mode: Off
** end
```

### xStatus SIP 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 CallForward URI
*s SIP CallForward URI: ""
** end
```

### xStatus SIP Mailbox MessagesWaiting

Indicates how many new messages are in the mailbox.

**Value space of the result returned:**

Integer

**Example:**

```
xStatus SIP Mailbox MessagesWaiting
*s SIP Mailbox MessagesWaiting: 0
** end
```

### xStatus SIP Mailbox URI

Returns the URI for your SIP mailbox.

**Value space of the result returned:**

String

**Example:**

```
xStatus SIP Mailbox URI
*s SIP Mailbox URI: "12345678"
** end
```

### xStatus SIP Proxy [n] Address

Shows the address of the SIP Proxy that the system communicates with.

**Value space of the result returned:**

String

**Example:**

```
xStatus SIP Proxy 1 Address
*s SIP Proxy 1 Address: "192.0.2.50"
** end
```

### xStatus SIP Proxy [n] 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 Proxy 1 Status
*s SIP Proxy 1 Status: Active
** end
```

### xStatus SIP Registration [n] Authentication

Shows which authentication mechanism is used when registering to the SIP Proxy Server.

Digest: Uses the Digest access authentication method, as specified by RFC 2069.

NTLM: Uses the NTLM authentication method, which is a Microsoft authentication protocol.

Off: No authentication mechanism is used.

**Value space of the result returned:**

Digest/Off

**Example:**

```
xStatus SIP Registration 1 Authentication
*s SIP Registration 1 Authentication: Off
** end
```

### xStatus SIP 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 Registration 1 Reason
*s SIP Registration 1 Reason: "404 Not Found"
** end
```

### xStatus SIP 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 Registration 1 Status
*s SIP Registration 1 Status: Registered
** end
```

### xStatus SIP Registration [n] URI

Shows the URI used for registration to the SIP Proxy server.

**Value space of the result returned:**

String

**Example:**

```
xStatus SIP Registration 1 URI
*s SIP Registration 1 URI: "firstname.lastname@company.com"
** end
```

### xStatus SIP Secure

Shows the encryption status of the signaling with the SIP Proxy server.

**Value space of the result returned:**

True/False

**Example:**

```
xStatus SIP Secure
*s SIP Secure: True
** end
```

### xStatus SIP 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 DefaultTransport not set to TLS) or certificate verification is switched off (SIP TlsVerify: Off. This setting is accessible through your products web interface).

**Value space of the result returned:**

True/False

**Example:**

```
xStatus SIP Verified
*s SIP Verified: False
** end
```

## Standby status

### xStatus Standby

Shows the top level overview of the standby status.

### xStatus Standby State

Shows whether the system is in standby mode or not.

Standby: The system is in standby state.

EnteringStandby: The system is entering the standby state.

Off: The system is not in standby.

#### Value space of the result returned:

Standby/EnteringStandby/Off

#### Example:

```
xStatus Standby State
*s Standby State: Off
** end
```

## SystemUnit status

### xStatus SystemUnit

Shows the top level overview of the system unit status.

### 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 Module SerialNumber

Shows the serial number of the hardware module in the codec.

#### Value space of the result returned:

String

#### Example:

```
xStatus SystemUnit Hardware Module SerialNumber
*s SystemUnit Hardware Module SerialNumber: "F9AA99A00090"
** end
```

### xStatus SystemUnit Hardware Monitoring Fan [n] Status

The feedback shows the speed (rpm) for the specified fan.

**Value space of the result returned:**

String

**Example:**

```
xStatus SystemUnit Hardware Monitoring Fan 1 Status
*s SystemUnit Hardware Monitoring Fan 1 Status: "locked on 1096 rpm"
** end
```

### xStatus SystemUnit Hardware Temperature

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 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 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 ProductId

Shows the product identity.

**Value space of the result returned:**

String

**Example:**

```
xStatus SystemUnit ProductId
*s SystemUnit ProductId: "Cisco TelePresence Codec SX80"
** end
```

### xStatus SystemUnit ProductPlatform

Shows the product platform.

**Value space of the result returned:**

String

**Example:**

```
xStatus SystemUnit ProductPlatform
*s SystemUnit ProductPlatform: "SX80"
** end
```

### xStatus SystemUnit ProductType

Shows the product type.

**Value space of the result returned:**

String

**Example:**

```
xStatus SystemUnit ProductType
*s SystemUnit ProductType: "Cisco Codec"
** end
```

### xStatus SystemUnit Software DisplayName

Shows the name of the software that is installed on the codec, as it is displayed in the UI.

**Value space of the result returned:**

String

**Example:**

```
xStatus SystemUnit Software DisplayName
*s SystemUnit Software DisplayName: "Display Name"
** end
```

### xStatus SystemUnit Software Name

Shows the name of the software that is installed on the codec.

**Value space of the result returned:**

String

**Example:**

```
xStatus SystemUnit Software Name
*s SystemUnit Software Name: "s52010"
** end
```

### xStatus SystemUnit Software OptionKeys RemoteMonitoring

Shows whether the system has the remote monitoring option key installed. Remote monitoring option key enables snapshots from the web interface, and from a remote paired Touch 10.

**Value space of the result returned:**

False/True

**Example:**

```
xStatus SystemUnit Software OptionKeys RemoteMonitoring
*s SystemUnit Software OptionKeys RemoteMonitoring: "true"
** 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: "2015-05-05"
** 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: "CE8.0.0"
** 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 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 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 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
```

## Time status

### xStatus Time

Shows the top level overview of the time status.

### 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

Shows the top level overview of the video status.

### xStatus UserInterface ContactInfo ContactMethod [n] Number

Returns the system's active contact information. This address is used to reach this endpoint.

**Value space of the result returned:**

String

**Example:**

```
xStatus UserInterface ContactInfo ContactMethod Number
*s UserInterface ContactInfo ContactMethod 1 Number: "12345678"
** end
```

### xStatus UserInterface ContactInfo Name

Returns the system's active contact name. The result depends on which protocol, if any, the system is registered on. The automatically set contact name may have been overridden with the command xConfiguration UserInterface ContactInfo Type. This results in a diagnostics warning about contact mismatch.

**Value space of the result returned:**

String

**Example:**

```
xStatus UserInterface ContactInfo Name
*s UserInterface ContactInfo Name: "MySystem"
** end
```

### xStatus UserInterface Extensions Widget [n] WidgetId

Returns the unique identifier for a widget. Widgets are used in custom panels, for example in-room controls, that are added to the user interface.

**Value space of the result returned:**

String

### xStatus UserInterface Extensions Widget [n] Value

Returns the value for a widget. Widgets are used in custom panels, for example in-room controls, that are added to the user interface. The value space depends on the widget type. The value is an empty string until a value is set with the UserInterface Extensions Widget SetValue command.

**Value space of the result returned:**

String

## Video status

### xStatus Video

Shows the top level overview of the video status.

### xStatus Video ActiveSpeaker PIPPosition

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 ActiveSpeaker PIPPosition
*s Video PIP ActiveSpeaker Position: UpperCenter
** end
```

### xStatus Video Input

Shows the top level overview of the video input status.

### xStatus Video Input Connector [n] Connected

Shows whether is something is 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 identifier 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 MainVideoSource

Returns the local video input currently used as the main source. The main video source is set with the xConfiguration Video DefaultMainSource 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:**

Integer

**Example:**

```
xStatus Video Input Source 1 ConnectorId
*s Video Input Source 1 ConnectorId: 1
** end
```

### xStatus Video Input Source [n] 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 FormatStatus
*s Video Input Source 1 Resolution FormatStatus: Ok
** end
```

### xStatus Video Input Source [n] 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 FormatType
*s Video Input Source 1 Resolution FormatType: Digital
** end
```

### 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 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 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 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 Layout LayoutFamily Local

Shows the local screen layout family.

**Value space of the result returned:**

String

**Example:**

```
xStatus Video Layout LayoutFamily Local
*s Video Layout LayoutFamily Local: "equal"
** end
```

### xStatus Video Layout LayoutFamily Remote

Shows the remote screen layout family.

**Value space of the result returned:**

String

**Example:**

```
xStatus Video Layout LayoutFamily Remote
*s Video Layout LayoutFamily Remote: "equal"
** end
```

### 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/TriplePresentationOnly/Quadruple

**Example:**

```
xStatus Video Monitors
*s Video Monitors: Single
** 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] 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 Connector [n] CEC Mode set to on.

**Value space of the result returned:**

Unknown/TV/Reserved/Recorder/Tuner/Playback/Audio

**Example:**

```
xStatus Video Output Connector 1 ConnectedDevice CEC DeviceType
*s Video Output Connector 1 ConnectedDevice CEC DeviceType: TV
** end
```

### xStatus Video Output Connector [n] 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 Connector 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 Connector 1 ConnectedDevice CEC PowerControl
*s Video Output Connector 1 ConnectedDevice CEC PowerControl: Ok
** end
```

### xStatus Video Output Connector [n] ConnectedDevice CEC PowerStatus

Shows the state of 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 Connector 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 Connector 1 ConnectedDevice CEC PowerStatus
*s Video Output Connector 1 ConnectedDevice CEC PowerStatus: Ok
** end
```

### xStatus Video Output Connector [n] 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 Connector 1 ConnectedDevice Name
*s Video Output Connector 1 ConnectedDevice Name: "G2420HDBL"
** end
```

### xStatus Video Output Connector [n] 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 Connector 1 ConnectedDevice PreferredFormat
*s Video Output Connector 1 ConnectedDevice PreferredFormat: "1920x1080@60Hz"
** 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 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] 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] 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 Presentation PIPPosition

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 Presentation PIPPosition
*s Video PIP Presentation Position: CenterLeft
** 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 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 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 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
```



## 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 SystemUnit Boot.

Use the codec's web interface 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 are 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 ?
license
network
ntp
passwd
pki
rootsettings
securitysettings
securitystatus
selectsw
sudo
whoami
touch
```

To see the usage of the commands add a question mark after the command.

### Example:

```
systemtools network ?
usage: network ping <hostname> | traceroute
<hostname> | netstat | addr | ifconfig
```

### 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>: The name of the license file.

### systemtools network ping <hostname>

Network debug command.

<hostname>: The IP address or URL of the host.

### systemtools network traceroute <hostname>

Network debug command.

<hostname>: The IP address or URL of the host.

### systemtools network netstat

Network debug command.

### systemtools network addr

Check the systems IP address.

### 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>: 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 compliant.

### systemtools securitysettings default

Revert to default security settings.

### systemtools securitysettings ask

Query for the separate configurations. When issuing this command you can see each policy separately.

- Press enter to keep the current value.
- Enter a number and press enter to change the given policy.
- The default value “0” indicates no restrictions.

Max failed login attempts [0]?

- Number of failed logins until a user is set inactive.

Suspend-time after max failed login attempts (minutes) [0]?

- Number of minutes the user is set inactive after maximum failed login attempts have been exceeded.

Max simultaneous sessions total [0]?

- Maximum number of users that can be logged in simultaneous to web and maximum number of users that can be logged in simultaneous to ssh/Telnet.

Max simultaneous sessions per user [0]?

- Maximum number of simultaneous sessions per user.

Number of passwords to remember [0]?

- Number of previous passwords that the new password must differ from.

Number of PINs to remember [0]?

- Number of previous PINs that the new PIN must differ from.

Maximum time between password renewals (days) [0]?

- If the user has not changed the password within the renewal time the user will be set inactive.

Minimum time between password renewals (hours) [0]?

- The user can only change password once within this limit.

Maximum time between PIN renewals (days) [0]?

- If the user has not changed the PIN within the renewal time the user will be set inactive.

Minimum time between PIN renewals (hours) [0]?

- The user can only change PIN once within this limit.

Maximum time between logins (days) [0]?

- If the user has not logged in within this limit the user will be set inactive.

Max consecutive equal digits in PINs [0]?

- Maximum consecutive equal digits in PINs.

Minimum number of digits in PINs [0]?

- Minimum number of digits in PINs.

Maximum number of digits in PINs [0]?

- Maximum number of digits in PINs.

Max consecutive identical characters in passwords [0]?

- Maximum consecutive identical characters in passwords.

Minimum number of characters in passwords [0]?

- Minimum number of characters in passwords.

Maximum number of characters in passwords [0]?

- Maximum 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>*: The name of the sudo user.

*<password>*: 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.

## About disconnect cause types

The following parameters are logged when a call is disconnected. The disconnect cause types are used in disconnect events (xEvent).

CauseValue	Proprietary. We recommend 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
```

## User documentation on the Cisco web site

User documentation for the Cisco TelePresence products is available at

▶ <http://www.cisco.com/go/telepresence/docs>

Choose a product category in the right pane until you find the correct product. This is the path you have to follow:

*Collaboration Endpoints >*

*Collaboration Desk Endpoints*

Alternatively, use the following short-link to find the documentation: ▶ <http://www.cisco.com/go/dx-docs>

**The documents are organized in the following categories:**

***Install and Upgrade > Install and Upgrade Guides***

- *Installation guides:* How to install the product
- *Getting started guide:* Initial configurations required to get the system up and running
- *RCSI guide:* Regulatory compliance and safety information

***Maintain and Operate > Maintain and Operate Guides***

- *Getting started guide:* Initial configurations required to get the system up and running
- *Administrator guide:* Information required to administer your product

***Maintain and Operate > End-User Guides***

- *User guides:* How to use the product
- *Quick reference guides:* How to use the product

***Reference Guides | Command references***

- *API reference guides:* Reference guide for the Application Programmer Interface (API)

***Reference Guides > Technical References***

- *CAD drawings:* 2D CAD drawings with measurements

***Software Downloads, Release and General Information > Licensing Information***

- *Open source documentation:* Licenses and notices for open source software used in this product

***Software Downloads, Release and General Information > Release Notes***

- *Software release notes*

### 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](http://www.cisco.com/go/offices).

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](http://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](http://www.cisco.com/web/siteassets/contacts)

Corporate Headquarters

Cisco Systems, Inc.

170 West Tasman Dr.

San Jose, CA 95134 USA