Software version TC5.1
MARCH 2012

Application Programmer Interface (API) Reference Guide

Cisco TelePresence System Codec C60/C40

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/telepresence/docs

Table of Contents

Appendices

Chapter 1

Introduction

About this guide

This guide will introduce you to the Application Programmer Interface (API) for the TC software in general, and serve as a reference guide for the command line commands for the Codec C Series listed below:

- Cisco TelePresence System Codec C60
- Cisco TelePresence System Codec C40

User documentation

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

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

Download the user documentation

Go to: ▶ http://www.cisco.com/go/telepresence/docs

- in the right pane, select:

- ***TelePresence Peripherals*** for the PrecisionHD cameras, microphones, Touch unit, and remote controls.
- ***TelePresence Solutions Platform*** for the Codec C Series and Quick Set C20.

What's new in this version

This section provides an overview of the new and changed advanced settings and new features in the TC5.1 software version.

Software release notes

For a complete overview of the news and changes, we recommend reading the Software Release Notes (TC5).

Go to: http://www.cisco.com/en/US/products/ps11422/tsd_products_support_series_home.html

Software download

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

New features and improvements

Extend meeting notification on screen when using Touch

A notification is shown on screen 5 minutes before a scheduled meeting is about to end. The same notification is displayed on the Touch controller and allows the user to extend the meeting by pressing Yes.

Web snapshots from remote sites

Web snapshots of the remote incoming video are available for unencrypted calls.

Web interface enhancements

Software versions with valid release keys are listed on the Software Upgrade page.

The system can be reset to factory default settings from the Maintenance tab.

Facility services

The API supports five faciity services.

The Touch controller uses one of them. If configured, a help button (?) will appear in the top banner. By pressing the help button, a call button will appear, serving as speed dial to a facility service, e.g. Help desk, Taxi or Reception. Current calls will be placed on hold when calling the facility service.

Controlling additional cameras from Touch

A drop-down menu allows selecting one of the connected cameras as main video source. The camera controls will operate the camera that is selected as main video source.

New languages supported on the Touch controller

- Danish
- Norwegian
- Spanish

xConfiguration changes

New configuration

xConfiguration Conference [1..1] Multipoint Mode

xConfiguration Experimental Conference [1..1] MultiStream Mode

xConfiguration Experimental Conference [1..1] MultiStream InputCount

xConfiguration Experimental Conference [1..1] MultiStream OutputCount

xConfiguration Experimental Conference [1..1] MultiStream Stream [1..4] Source

xConfiguration FacilityServices Service [1..5] Type

xConfiguration FacilityServices Service [1..5] Name

xConfiguration FacilityServices Service [1..5] Number

xConfiguration FacilityServices Service [1..5] CallType

Configurations that are removed

xConfiguration Experimental PacketOverloadHandling WhenDetected

Configurations that are modified

xConfiguration Conference DoNotDisturb Mode

OLD: <On/Off>

NEW: <On/Off/Timed>

xConfiguration Video Output HDMI [1] Resolution

OLD: <Auto/640_480_60/800_600_60/1024_768_60/1280_1024_60/1280_720_50/1280_720_60/1920_1080_50/1920_1080_60/1280_768_60/1360_768_60/1366_768_60/1600_1200_60/1920_1200_60>

NEW: <Auto/640_480_60/800_600_60/1024_768_60/1280_1024_60/1280_720_50/1280_720_60/1920_1080_50/1920_1080_60/1280_768_60/1360_768_60/1366_768_60/1600_1200_60/1680_1050_60/1920_1200_60>

xConfiguration Video Output DVI [2] Resolution

OLD: <Auto/640_480_60/800_600_60/1024_768_60/1280_1024_60/1280_720_50/1280_720_60/1920_1080_50/1920_1080_60/1280_768_60/1360_768_60/1366_768_60/1600_1200_60/1920_1200_60>

NEW: <Auto/640_480_60/800_600_60/1024_768_60/1280_1024_60/1280_720_50/1280_720_60/1920_1080_50/1920_1080_60/1280_768_60/1360_768_60/1366_768_60/1600_1200_60/1680_1050_60/1920_1200_60>

xCommand changes

New commands

xCommand Audio Vumeter StopAll

xCommand DoNotDisturb Activate

xCommand DoNotDisturb Dectivate

xCommand FacilityService Dial

xCommand Phonebook ContactMethod Modify

xCommand SystemUnit Notifications RemoveAll

xCommand SystemUnit ReleaseKey List

Commands that are removed

xCommand Provisioning CancelUpgrade

xCommand Provisioning CompleteUpgrade

Commands that are modified

xCommand Boot

ADDED: Action: <Restart/Shutdown>

xCommand Experimental TakeWebSnapshot

OLD: SourceType: <localInput/localMain/localPresentation>

NEW: SourceType: <localInput/localMain/localPresentation/presentation/remoteMain>

xCommand Phonebook Contact Add

ADDED: CallType: <Audio/Video>

xCommand Phonebook ContactMethod Add

ADDED: CallType: <Audio/Video>

xCommand Provisioning StartUpgrade

REMOVED: AutoComplete

xCommand SystemUnit FactoryReset

ADDED: TrailingAction: <NoAction/Restart/Shutdown>

xStatus changes

New commands

xStatus Conference Multipoint Mode

xStatus Conference DoNotDisturb

xStatus Notifications

Commands that are removed

xStatus SystemUnitHardware AudioBoard Identifier

xStatus SystemUnitHardware AudioBoard SerialNumber

Chapter 2

About the API

About the API

Basic Principles

The heart of the API is the API-Engine. This is where all information is stored and processed.

The API-engine can be accessed by an easy-to-use Command Line Interface called ***XACLI*** using RS-232, ***Telnet*** or ***SSH***, or by the XML API Service (***TXAS***) over HTTP/HTTPS.

Working with the API-engine is very similar to working with catalogues and files on a computer. All information is stored in a hierarchic tree structure which is accessible from different interfaces.

- When accessing the API-engine using XACLI (RS-232, Telnet or SSH), the information is formatted in a proprietary Command Line style or in XML formatting.
- When accessing the API-engine using the TXAS interface (HTTP/HTTPS), XML formatting is supported.

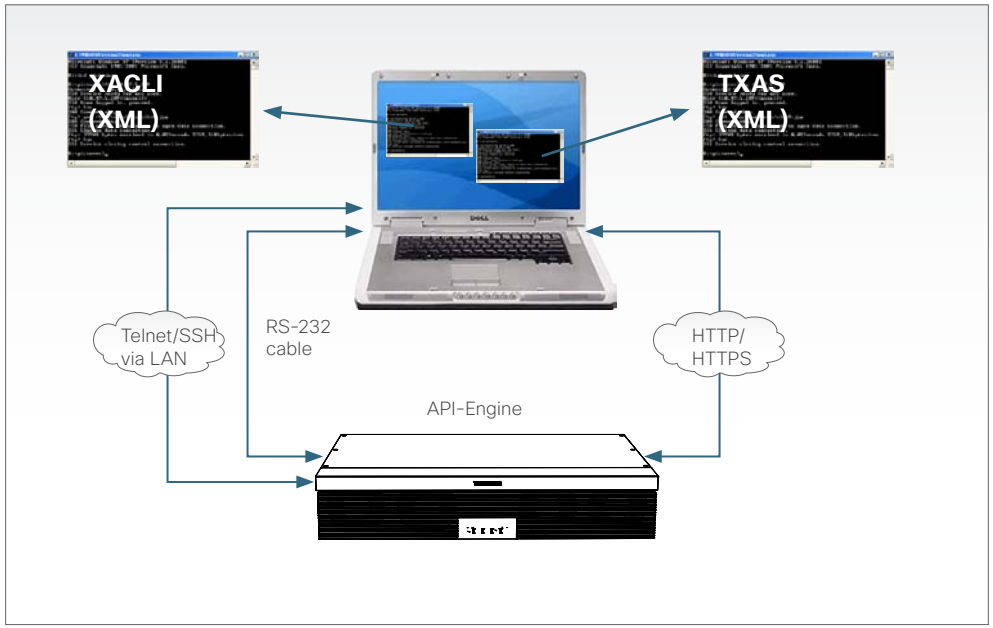
This is similar to viewing files on a computer. Accessing catalogues on a Windows computer using the Command Prompt gives a different view than using Windows Explorer, but the information is the same.

About Telnet

Telnet is disabled by default. Before connecting to the codec using Telnet you will need to enable the interface via either RS-232 or SSH.

The following command can be set from the Administrator settings menu or from the API command interface:

- `xConfiguration NetworkServices Telnet Mode: On`



The API-Engine

The API-Engine is optimized for easy, yet advanced, machine-machine interaction between a Cisco system and an external control application.

The main features can be summarized to:

1. Structuring of information
2. Addressing using XPath (XML Path Language) or SimplePath
3. Feedback

Structuring of Information

An application programming interface (API) can be seen as a gate where information is exchanged between two systems – a control application and a target system.

The control application transmits instructions to the target system, while the target system supplies information about how these instructions are executed, in addition to other system related information.

Consequently, the exchange of information can be divided into:

1. Information flowing from target. This we call READ information (R). The (R) should not be confused with the (r) used to indicate required parameters in the Commands tables.
2. Information flowing to target. This we call WRITE information (W).

Main types of information

- READ information (R)
- WRITE information (W)
- READ/WRITE information (RW)

(R) READ information. This is Status Information about the system and system processes, i.e. information generated by the system.

Typical examples include: status about ongoing calls, network status, conference status etc. All status information is structured in a hierarchy, making up a database constantly being updated by the system to reflect process changes.

(W) WRITE information. This is Command information the user/control application supply to initiate an action.

Typical examples include: instructing the system to place a call, adjust volume, disconnect a call etc.

A command is usually followed by a set of parameters to specify how the given action is to be executed.

(RW) READ/WRITE information. This is Configuration Information defining system settings. This information can both be supplied and read by the user/control application. Typical examples include: default call rate, baud rate of a serial port, enabling/disabling of various features etc.

All configuration information is structured in a hierarchy making up a database of system settings. But for the Configuration information, the data in the database can only be updated by the user/control application.

Addressing Using XPath or SimplePath

To address information in the hierarchic structure of Status and Configuration information, the Cisco systems support abbreviated XML Path Language (XPath) and a proprietary notation called SimplePath (only available using XACLI).

This allows the user/control application to address everything from a single element of data (for example the call rate of a specific call) to larger parts of the hierarchy (for example all information available for a given call).

Using XPath

Addressing the 1st DNS Server Address of the 1st Network:

Each level is separated with a slash ('/'). Item numbers are added in brackets after the element name:

- Network[1]/DNS Server[1]/Address

Example:

```
xConfiguration Network[1]/DNS Server[1]/Address
*c xConfiguration Network 1 DNS Server 1 Address:
"test"
OK
```

Using SimplePath

Addressing the 1st DNS Server Address of the 1st Network:

Both levels and item numbers are separated with white spaces:

- Network 1 DNS Server 1 Address

Example:

```
xConfiguration Network 1 DNS Server 1 Address
*c xConfiguration Network 1 DNS Server 1 Address:
"test"
OK
```

Feedback

Feedback is an extremely powerful feature where the Cisco system actively returns updated status and configuration information to the user/control application whenever changes occur.

The user/control application can specify what parts of the status and configuration hierarchies it wants to monitor by using XPath. The user/control application can thereby limit the amount of information it receives from the target system to only those parts being of interest for the given application. This will also reduce the load on the link connecting the systems.

Feedback is supported on both XACLI (RS-232/Telnet/SSH) and TXAS (HTTP/HTTPS) simultaneously.

The system uses SimplePath when presenting configurations.

XPath and SimplePath are described thoroughly later in this section of the manual.

The structuring of information together with XPath and SimplePath for addressing, makes up powerful features as the ability to search and setting of multiple instances of a configuration.

Connecting to the codec

Accessing XACLI

XACLI can be accessed through Telnet and SSH via the LAN interface or through the COM port by connecting a serial cable to the serial interface connector, referred to as the COM port.

The COM port (RS-232) is a 9-pin, female, D-sub connector located on the back of the Codec C-Series. The connector i marked with the text: *Camera Control*.

The port is configured as a DCE (Data Communications Equipment). The COM port (RS-232) is default set to 38400 baud, 8 data bits, none parity and 1 stop bit from factory. The port may also be referred to as the Data port.

Telnet/SSH login

Telnet is by default disabled. This can be changed with a configuration command: xConfiguration NetworkServices Telnet Mode: On/Off

- `xConfiguration NetworkServices Telnet Mode: On`

Telnet/SSH login

- User name is: `admin`
- The default passworrd is blank.

Serial port login

The serial port is password protected by default. The password protection may be configured.

- User name is: `admin`
- The default passworrd is blank.

Serial port configurations

On the serial port the baud rate and password protection may be configured.

The configuration command for the baud rate is:

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

- `xConfiguration SerialPort BaudRate: 38400`

The configuration command for login required is:

xConfiguration SerialPort LoginRequired: <On/Off>

- `xConfiguration SerialPort LoginRequired: On`

Reboot

The system requires a reboot for the changes to baud rate and password protection to take effect.

NOTE: When system boots up the baud rate of the boot messages is 38400 regardless of the baud rate set in the codec application.

Hardware & Cabling (RS-232)

The pin outs for the RS-232 are defined in the tables to the right. Observe that the DTE (Data Terminal Equipment), could be a PC or any other device capable of serial communication.

Cable. A straight-through cable should be used between the RS-232 port and the DTE. The lower table shows the recommended cable-wiring scheme when connecting the Codec C-Series to a PC through RS-232.

DTR and RTS are ignored. DSR, CD, and CTS are always asserted, while RI is not used.

Troubleshooting (RS-232)

If communication cannot be established between the PC/terminal and the Codec data port, the following should be checked:

1. Verify that the serial cable is a straight-through 9-pin to 9-pin cable.
2. Confirm that the configuration of the PC/terminal's serial RS-232 port is identical to the configuration of the RS-232 port.
3. Verify that the PC/terminal's serial RS-232 port is working properly by connecting it back-to-back to another PC/terminal and send characters in both directions.

COM port (RS-232)

Pin	*Signal name*	*Direction*
1	Carrier detect, CD	From DCE
2	Receive data, RXD	From DCE
3	Transmit data, TXD	To DCE
4	Data terminal ready, DTR	From DCE
5	Signal GND	
6	Data set ready, DSR	From DCE
7	Ready to send, RTS	To DCE
8	Clear to send, CTS	From DCE
9	Ring indicator, RI	From DCE

Cable wiring (RS-232) DCE <-> PC

DCE 9 pin	*Direction*	*PC DTE, 9 pin*
1 CD	–>	1 CD
2 RD	–>	2 RD
3 TD	<–	3 TD
4 DTR	<–	4 DTR
5 GND	<–>	5 GND
6 DSR	–>	6 DSR
7 RTS	<–	7 RTS
8 CTS	–>	8 CTS
9 RI	–>	9 RI

Value types and formats

The system supports the following value types:

- Integer values
- Literal values
- String values

Strings can have rules that further specify their format and length. Integer input may have a limited valid range.

Formats for value types
Integer values:<x..y>
Defines the valid range for an integer input. x = min value, y = max value.
<1..100>
Literal values: <X/Y/../Z>
Defines the possible values for a given configuration.
<On/Off/Auto>
String values: <S: x, y>
Defines that the valid input for this configuration is a String with minimum length x and maximum length of y characters.
<S: 0, 49>

User commands

By typing ? or help after connecting to the Cisco TelePresence System Codec C Series using RS-232/Telnet/SSH, the system will list all supported root commands.

Bye

The bye command will close the command line interface.

Echo <on/off>

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

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

The oher commands

The other user commands are described in the following pages.

```
?

- User Commands -

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

OK
```

Main type of commands

The XACLI is divided into three main types of commands, reflecting the information types supported by the API Engine.

The main types are:

Configuration type commands

Status type commands

Command type commands

Configuration type commands

Configuration type commands defines the system settings. Configuration commands are either supplied or read by the user.

System settings made by Configuration type commands are persitant over a reboot.

Example: Set default call rate, baud rate of a serial port and enabling/disabling of various features etc.

The Configuration commands are structured in a hierarchy, making up a database of system settings.

Supported Configuration type commands:

- xConfiguration

Command type commands

Command type commands instructs the system to perform an action. xCommand commands are supplied by the user.

Actions performed by Command type commands are not persitant over a reboot.

Example: instructing the system to place a call, assign floor to a specific site, disconnect a call etc.

A xCommand command is usually followed by a set of parameters to specify how the given action is to be executed.

Supported Command type commands:

- xCommand

Status type commands

Status type commands returns information about the system and system processes. Status type commands are read by the user.

Example: Information generated by the system about ongoing calls, network status, conference status etc.

All status information is structured in a hierarchy, making up a database constantly being updated by the system to reflect system and process changes.

Supported Status-type commands:

- xStatus
- xHistory

Special commands

In addition to the above sets of commands, XACLI supports the following set of special commands:

Feedback type command

The Feedback commands are used to specify what parts of the configuration and status hierarchies to monitor. Feedback will only be issued on the RS-232/Telnet/SSH session for which it is specified. If connecting to the codec with multiple sessions, each session can define feedback individually.

Supported Feedback type commands.

- xFeedback
- xEvent

Preferences type command

The Preference type command is used to set various preferences for the RS-232/Telnet/SSH sessions. Each session can define preferences individually. **IMPORTANT!** This command has various settings to define the formatting of the XACLI output. It is therefore important to define settings to match the parser used on the control system. XACLI is designed to make parsing of data from the Codec C-Series very simple.

More on this can be found in xpreferences.

- xPreferences

About xConfiguration

The xConfiguration type commands defines the system settings and are either supplied or read by the user. The xConfigurations commands are organized in a hierarchic tree structure.

To get an overview of accessible top-level configuration elements within the xConfiguration commands, enter ? or help after the xConfiguration command:

- `xConfiguration ?`
- `xConfiguration help`

To get an overview of all supported xConfiguration commands with the corresponding value space, enter ?? after the xConfiguration command:

- `xConfiguration ??`

When issuing a xConfiguration command, the command consists of three parts:

1. The type of command: xConfiguration
2. The path: An address expression, terminated by a colon
3. The value: A value type

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

The type | The path | The value

```
xConfiguration ?

- User Configurations -
 Audio                  H323                   Provisioning           Standby
 Cameras                Network                RTP                    SystemUnit
 Conference             NetworkPort            Security               Time
 Experimental           NetworkServices        SerialPort             UserInterface
 FacilityService        Phonebook              SIP                    Video


OK
```

```
xConfiguration ??

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

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

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

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

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

*? xConfiguration Audio Input Microphone [1..2] Type: <Microphone/Line>

*? xConfiguration Audio Input Microphone [1..2] Mode: <Off/On>

*? xConfiguration Audio Input Microphone [1..2] Level: <0..24>

*? xConfiguration Audio Input Microphone [1..2] EchoControl Mode: <Off/On>
.
.
.
OK
```

xConfiguration operations

The xConfiguration type commands defines system settings and are either supplied or read by the user.

Return result parameters

Three operations can be performed on xConfiguration:

Configuration Help

- The value space for the configuration is returned.

Configuration Read

- *c is used when returning the result of a read query.

Configuration Write

- No return result parameter for configuration set (write).
- Writes this value to the setting defined by the path.

xConfiguration Help

To get help on a system setting you can use a help query. Enter the path followed by **?** or `help`.

- `xConfiguration H323 Profile 1 Gatekeeper Discovery ?`
  Returns information about the setting defined by the path.
- `xConfiguration H323 Profile 1 Gatekeeper Discovery help`
  As above.

Example with xConfiguration Help:

To get help on xConfiguration, type ? or help after the configuration path (address expression):

xConfiguration <address expression> ?

```
xConfiguration H323 Profile 1 Gatekeeper Discovery ?
*? xConfiguration H323 Profile [1..1] Gatekeeper Discovery: <Manual/Auto>

OK
```

xConfiguration Read

When reading a value you will use the configuration read. The level of details is defined by the path:

- `xConfiguration H323 Profile 1 Gatekeeper Discovery`
  Returns the current value of the setting defined by the path.

Example with xConfiguration Read:

To read configurations from the system just type the root command (xConfiguration) followed by the path (address expression):

xConfiguration <address expression>

```
xConfiguration H323 Profile 1 Gatekeeper Discovery
*c xConfiguration H323 Profile 1 Gatekeeper Discovery: Manual
** end

OK
```

xConfiguration Write

When defining a system setting you will use the configuration write. The structure is described by the example to the right.

- `xConfiguration H323 Profile 1 Gatekeeper Discovery: Auto`
  Writes this value to the setting defined by the path.

Example with xConfiguration Write:

To issue a command type a root command (xConfiguration) followed by a valid path (address expression). The path must be terminated with a colon before the value is added:

xConfiguration <address expression>: <value>

```
xConfiguration H323 Profile 1 Gatekeeper Discovery: Auto
** end

OK
```

About xCommand

The xCommand type commands instructs the system to perform an action. xCommand type commands are supplied by the user.

To get an overview of the supported xCommand type commands, type ? or help after the xCommand:

- `xCommand ?`
- `xCommand help`

To get an overview of all supported xCommand commands with the corresponding value space, enter ?? after the xCommand:

- `xCommand ??`

```
xCommand ?

- User Commands -
 Audio                  Conference             Key                    SStringSend
 Bookings               Dial                   Message                Standby
 Boot                   DTMFSend               Phonebook              SystemUnit
 Call                   Experimental           Presentation           TStringSend
 CallLog                FacilityService        Preset                 Video
 CamCtrlPip             FarEndControl          Provisioning
 Camera                 HttpFeedback           Security

OK
```

When you type a command and ? or help a list of the available parameters will show. Required parameters are identified by an (r) behind the parameter name.

```
xCommand ??

xCommand Audio Microphones Mute
xCommand Audio Microphones Unmute
xCommand Audio Sound Play
    Sound(r): <Busy/CallWaiting/Dial/KeyTone/Ringing/SpecialInfo/TelephoneCall/VideoCall>
    Loop: <On/Off>
xCommand Audio Sound Stop
xCommand Audio Vumeter Start
    ConnectorType(r): <HDMI/Line/Microphone>
    ConnectorId(r): <1..4>
xCommand Audio Vumeter Stop
    ConnectorType(r): <HDMI/Line/Microphone>
    ConnectorId(r): <1..4>
xCommand Audio Vumeter StopAll
xCommand Audio Setup Clear
.
.
.
OK
```

xCommand operations

The xCommand type commands are used to instruct the system to perform a given action.

Return result parameters

The following operations can be performed on xCommand:

Command Help

- A list of return values with value space is returned.

Command Write

- *r is used when returning the result of a write command.

xCommand Help

To get help on a setting you can use a help query. Enter the path followed by **?** or `help`.

- `xCommand Dial ?`
  Returns a set of return values. See the example to the right.
- `xCommand Dial help`
  As above.

Example with xCommand Help

To get help on xCommand, type ? or help after the command path (address expression):

xCommand <address expression> ?

```
xCommand Dial ?
xCommand Dial
    Number(r): <S: 0, 255>
    Protocol: <H323/Sip>
    CallRate: <64..6000>
    CallType: <Audio/Video>
    BookingId: <S: 0, 255>

OK
```

xCommand Write

When issuing a command, the system will return a set of return values. The structure is described by the example to the right.

- `xCommand Dial Number: 12345`
  Issues the command and gives a response. See the example to the right.

The response will by default be on the same format as the standard XACLI Status format. The XML status format is also supported.

You can read more about XML in the xPreferences section.

Example with xCommand Write

Dial a number with only the required parameter:

```
xCommand Dial Number: 95458458

OK
*r DialResult (status=OK):
    CallId: 2
    ConferenceId: 1
** end
```

About xStatus

The xStatus type commands returns information about the system and system processes. Status type commands are read by the user.

All status information is structured in a hierarchy, making up a database constantly being updated by the system to reflect system and process changes.

To get an overview of the supported xStatus type commands, type ? or help after the xStatus:

- `xStatus ?`
- `xStatus help`

```
xStatus ?

- Status -

 Audio                    H323                     Security
 Call[...]                HttpFeedback[1..4]       SIP
 Camera[1..7]             MediaChannels            Standby
 Conference               Network                  SystemUnit
 Diagnostics              Preset[1..15]            Time
 Experimental             Provisioning             Video

OK
```

Return result parameters

The following operation can be performed on xStatus commands:

xStatus Read

- *s is used when returning the result of xStatus read query.

Quering status information

The xStatus type commands returns information about the system and system processes. You can query all information or just some of it.

To address status information enter the xStatus command followed by an address expression (XPath or SimplePath).

You can set up the xStatus read command to address all information or just some of it, see the examples to the right for illustrations.

Address status information with xStatus

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

xStatus <address expression>

Example 1: Query all ongoing Call information:

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

OK
```

Example 2: Query the protocol for a call:

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

OK
```

About xHistory

The xHistory type commands returns information about what has happened on the system. History type commands are read by the user.

All history information is structured in a hierarchy, making up a database constantly being updated by the system to reflect system and process changes.

To get an overview of the supported xHistory type commands, type ? or help after the xHistory:

- `xHistory ?`
- `xHistory help`

```
xHistory ?

- History -

   CallLogs

OK
```

Return result parameters

xHistory Log

- *h is used when returning the result of xHistory log query

Example with xHistory CallLogs

```
xhistory
*h xHistory CallLogs Call 1 CallId: 13
*h xHistory CallLogs Call 1 Protocol: "h323"
*h xHistory CallLogs Call 1 Direction: Incoming
*h xHistory CallLogs Call 1 CallType: Video
*h xHistory CallLogs Call 1 RemoteNumber: "h323:firstname.lastname.office@company.com"
*h xHistory CallLogs Call 1 CallbackNumber: "h323:firstname.lastname.office@company.com"
*h xHistory CallLogs Call 1 DisplayName: "firstname.lastname@company.com"
*h xHistory CallLogs Call 1 CallRate: 768
*h xHistory CallLogs Call 1 DisconnectCauseValue: 2
*h xHistory CallLogs Call 1 DisconnectCause: "Normal"
*h xHistory CallLogs Call 1 DisconnectCauseType: RemoteDisconnenct
*h xHistory CallLogs Call 1 DisconnectCauseCode: 16
*h xHistory CallLogs Call 1 DisconnectCauseOrigin: Q850
*h xHistory CallLogs Call 1 StartTime: "2012/02/14 11:04:14"
*h xHistory CallLogs Call 1 Duration: 184
*h xHistory CallLogs Call 1 Encryption: "None"
*h xHistory CallLogs Call 1 BookingId: ""
...
*h xHistory CallLogs Recent 6 CounterMissed: 0
*h xHistory CallLogs Recent 6 Counter: 3
...
*h xHistory CallLogs Outgoing 30 Counter: 1
...
*h xHistory CallLogs Received 40 Counter: 1
...
*h xHistory CallLogs Missed 50 Counter: 2
*h xHistory CallLogs Missed 50 NewCounter: 0
** end
```

About xEvent

The xEvent type commands returns information about what events that are available for xFeedback.

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

- `xEvent ?`
- `xEvent ??`
- `xEvent help`

Return result parameters

- *es is used when returning the result of xEvent query

```
xEvent ?
  *es Event Audio
  *es Event Bookings
  *es Event CallDisconnect
  *es Event CallLog
  *es Event CallSuccessful
  *es Event CallTransfer
  *es Event CameraPresetActivated
  *es Event CameraPresetListUpdated
  *es Event FeccReceive
  *es Event IncomingCallIndication
  *es Event KeyDown
  *es Event KeyRelease
  *es Event KeyRepeat
  *es Event OutgoingCallIndication
  *es Event SStringReceive
  *es Event Shutdown
  *es Event SoftwareUpgrade
  *es Event Standby
  *es Event WebSnapshotTaken
  ** end
OK
```

Example with xEvent IncomingCallIndication

```
xevent IncomingCallIndication ??
*es Event IncomingCallIndication RemoteURI
*es Event IncomingCallIndication DisplayNameValue
*es Event IncomingCallIndication CallId
** end

OK
xFeedback register Event/IncomingCallIndication
** end

OK
*e IncomingCallIndication RemoteURI: "h323:559216" DisplayNameValue: "firstname.lastname@company.com"
CallId: 11
** end
```

About xFeedback

The xFeedback is a powerful feature on the Codec C90/C60/C40. It lets you subscribe to what you want to be notified about when changes occur on the system

- This can be configuration changes like someone changes the name of the system.
- It might be events like key press from the remote control.
- Or it can be changes to the state of the system, like a call connecting or disconnecting.

The xFeedback command is used to specify what parts of the configuration and status hierarchies to monitor, and will only be issued on the RS-232/Telnet/SSH for which it is specified.

If connecting to the codec with multiple sessions, each session can define feedback individually.

CAUTION: We discourage registering all status changes as this may give too much feedback information than the control systems are able to handle.

```
xFeedback ?

xFeedback help:
  xFeedback register XPathExpression[1..255]     - Registers feedback on expression XPathExpression.
  xFeedback deregister XPathExpression[1..255]   - Deregisters feedback if registered on XPathExpression.
  xFeedback deregisterall                        - Deregister all expressions.
  xFeedback list                                 - Generate list of currently registered XPathExpressions.
  xFeedback help                                 - Display this help text.
```

Example with xFeedback

```
xFeedback register Status/Audio
** end

OK
xFeedback register Event
** end

OK
xFeedback list
Event
Status/Audio
** end

OK
xFeedback deregister Event
** end

OK
xFeedback list
Status/Audio
** end

OK
```

About xPreferences

The xPreferences command is used to set various preferences for the RS-232/Telnet/SSH sessions.

Each session can define preferences individually.

IMPORTANT! This command has various settings to define the formatting of the XACLI output. It is therefore important to define settings to match the parser used on the control system. XACLI is designed to make parsing of data from the Codec C-Series very simple.

To get an overview of the supported xPreferences commands and their value space, type ? or help after the xPreferences:

- `xPreferences ?`
- `xPreferences help`

```
xPreferences ?

xpreferences usage:
 xpreferences outputmode <terminal/xml>
 xpreferences apiversion <1/2>
OK
```

The xPreferences output modes

- Terminal: Line based XACLI output for use with line based control systems
- XML: Pure XML output for use with control systems that understand XML. NOTE! This mode is to be considered experimental in version 1 of the software. Its format WILL change in next version.

The SystemTools commands

The systemtools command is used for administrative control of the codec and is only available from a command line interface.

Required parameters in angle brackets: <text>

Optional parameters in square brackets: [text]

To get an overview of the supported commands type "systemtools ?".

Example:

```
systemtools ?
authorizedkeys
license
network
pairing
passwd
pki
rootsettings
securitysettings
securitysatus

OK
```

To see the usage of the commands add a question mark after the command.

Example:

```
systemtools authorizedkeys ?
usage: authorizedkeys <add <method> <key>
[comment] | delete <id> | list | clear>

OK
```

systemtools authorizedkeys add <method> <key> [comment]

Add the ssh keys on the codec.

method(r): The encryption method used, which can be ssh-rsa or ssh-dss.

key(r): The public key as it is in the ssh public key file.

comment: Optional comment.

systemtools authorizedkeys delete <id>

Delete the given ssh key on the codec, defined by the id.

id(r): The id as as displayed in the authorized keys list.

systemtools authorizedkeys list

List the ssh keys on the codec.

systemtools authorizedkeys clear

Clear all ssh keys on the codec.

systemtools license list

Lists all the licenses for the codec.

systemtools license show <name>

Shows the content of a license file, define by the name.

name(r): The name of the license file.

systemtools network ping <hostname>

Network debug commands.

hostname(r): The IP address or URL of the host.

systemtools network traceroute <hostname>

Network debug commands.

hostname(r): The IP address or URL of the host.

systemtools network netstat

Network debug command.

systemtools pairing unpair

Remove association with Cisco TelePresence Touch controller.

systemtools passwd

Change the password for the logged in user.

systemtools pki list

Lists the codec certificate and CA list if they exist.

systemtools pki delete <cert-name>

Delete the codec certificate and CA list if they exist.

cert-name(r): The name of the certificate.

systemtools rootsettings get

Obtain the current setting for the systemtools rootsetting.

systemtools rootsettings on [password]

Command to control the root user availability.

Enable access to the system for the root user on all ports.

password: The root user password.

systemtools rootsettings serial [password]

Command to control the root user availability.

Enable access to the system for the root user on the serial port.

password: The root user password.

systemtools rootsettings off

Command to control the root user availability.

Disable access to the system for the root user on all ports.

systemtools rootsettings never

Command to control the root user availability.

NOTE! The root user is permanently turned off!

To get back the root user the system must be reset to factory defaults, ref. xCommand SystemUnit FactoryReset.

systemtools securitysettings jitc

Set up security requirements so they meet JITC.

Set password and PIN polices enforced on the codec.

systemtools securitysettings isjitc

Check if the current settings are JTIC compiant.

systemtools securitysettings default

Revert to default security settings.

systemtools securitysettings ask

Query for the separate configurations. When issuing this command you will see each policy separately.

- Press enter to keep the current value.
- Enter a number and press enter to change the given policy.
- The default value “0” indicates no restrictions.

Max failed login attempts [0]?

- Number of failed logins until a user is set inactive.

Suspend-time after max failed login attempts (minutes) [0]?

- Number of minutes the user is set inactive after maximum failed login attempts have been exceeded.

Max simultaneous sessions total [0]?

- Maximum number of users that can be logged in simultaneous to web and maximum number of users that can be logged in simultaneous to ssh/Telnet.

Max simultaneous sessions per user [0]?

- Maximum number of simultaneous sessions per user.

Number of passwords to remember [0]?

- Number of previous passwords that the new password must differ from.

Number of PINs to remember [0]?

- Number of previous PINs that the new PIN must differ from.

Maximum time between password renewals (days) [0]?

- If the user has not changed the password within the renewal time the user will be set inactive.

Minimum time between password renewals (hours) [0]?

- The user can only change password once within this limit.

Maximum time between PIN renewals (days) [0]?

- If the user has not changed the PIN within the renewal time the user will be set inactive.

Minimum time between PIN renewals (hours) [0]?

- The user can only change PIN once within this limit.

Maximum time between logins (days) [0]?

- If the user has not logged in within this limit the user will be set inactive.

Max consecutive equal digits in PINs [0]?

- Maximum consecutive equal digits in PINs.

Minimum number of digits in PINs [0]?

- Minimum number of digits in PINs.

Maximum number of digits in PINs [0]?

- Maximum number of digits in PINs.

Max consecutive identical characters in passwords [0]?

- Maximum consecutive identical characters in passwords.

Minimum number of characters in passwords [0]?

- Minimum number of characters in passwords.

Maxmum number of characters in passwords [0]?

- Maxmum number of characters in passwords.

Minimum number of lower-case letters in passwords [0]?

- Minimum number of lower-case letters in passwords.

Minimum number of upper-case letters in passwords [0]?

- Minimum number of upper-case letters in passwords.

Minimum number of numerical characters in passwords [0]?

- Minimum number of numerical characters in passwords.

Minimum number of special characters in passwords [0]?

- Minimum number of special characters in passwords.

Minimum number of character groups in passwords [0]?

- Minimum number of character groups in passwords.

Minimum number of character changed from previous password [0]?

- Minimum number of character changed from previous password.

systemtools securitystatus

Shows the security status for the codec.

XML API service

TXAS is a service provided by Cisco units for transmitting and receiving (transceiving) information encoded in XML format.

The API uses HTTP(S) as the transport mechanism and connects to the normal web port (80). TXAS can be accessed by bare-bone HTTP requests where URL's uniquely identifies the request.

Bare-bone HTTP/HTTPS Access

The bare-bone HTTP mode uses a unique URL to identify the specific request. The contents of the HTTP body will be a XML document (or part of it).

Bare-bone HTTP(S) access is accomplished by passing arguments in the query string (after '?' in URL) in a GET request, or using the "application/x-www-form-urlencoded" content-type method of POSTing form data (Each argument starts with a name '=' and a value, and every parameter separated with '&' (and opt NL).)

getxml

/getxml request returns an XML document based on the location parameter passed to the request. The elements (or complete document) matching the expression will be returned.

On Incorrect XPath expression, a <Fault> element with a <XPathError> element will be returned.

```
getxml
REQUEST:
  /getxml
PARAM:
  location = XPath expression
```

formputxml

This is most useful in a POST (to extend character limit of 255 of GET urls). It posts a Configuration or Command document to set the configurations or issue a command.

Like getxml, it has the data URL form-data encoded with one single parameter. The Content-Type of the document must be of type "application/x-www-form-urlencoded" and the body must be encoded accordingly (e.g. first line will be xmldoc=<then the document>).

```
formputxml
REQUEST:
  /formputxml
PARAM:
  xmldoc   = "an XML document of Configuration, Directory or Command"
```

putxml

Putxml is like formputxml+, put uses the complete BODY as argument (i.e. the content of the xmldoc parameter). The Content-type should be "text/xml" or "application/xml" (or "text/plain"), though no check at the moment. (Except for application/x-www-form-url encoded which will cause a failure).

```
putxml
REQUEST:
  /putxml
PARAM:
  HTTP BODY as argument
```

Chapter 3

Description of the xConfiguration commands

Description of the xConfiguration commands

In the following pages you will find a complete list of the xConfiguration commands. The examples shows 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/telepresence/docs

The Audio configuration

xConfiguration Audio Input HDMI [2] Mode

Determine if the audio channels on the HDMI input shall be enabled. The HDMI input 2 has two audio channels.

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Enable the audio channels on the HDMI input.

Off: Disable the audio channels on the HDMI input.

Example: `xConfiguration Audio Input HDMI 2 Mode: On`

xConfiguration Audio Input HDMI [2] Level

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

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

Requires user role: `ADMIN`

Value space: `<-24..0>`

Range: Select a value from -24 to 0 dB.

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

xConfiguration Audio Input HDMI [2] VideoAssociation MuteOnInactiveVideo

Enable association of a video source to an HDMI audio input.

Requires user role: `ADMIN`

Value space: `<On/Off>`

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

Off: No video source is associated.

Example: `xConfiguration Audio Input HDMI 2 VideoAssociation MuteOnInactiveVideo: Off`

xConfiguration Audio Input HDMI [2] VideoAssociation VideoInputSource

Select the associated video input source.

Requires user role: `ADMIN`

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

Range: Select one of the video input sources.

Example: `xConfiguration Audio Input HDMI 2 VideoAssociation VideoInputSource: 1`

xConfiguration Audio Input Line [1..2] Equalizer ID

Select the audio input line equalizer ID.

Requires user role: `ADMIN`

Value space: `<1..8>`

Range: Select EqualizerID 1 to 8.

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

xConfiguration Audio Input Line [1..2] Equalizer Mode

Set the audio input line equalizer mode.

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Enable the equalizer for the audio input line.

Off: No equalizer.

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

xConfiguration Audio Input Line [1..2] VideoAssociation MuteOnInactiveVideo

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

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

Off: No video source is associated.

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

xConfiguration Audio Input Line [1..2] VideoAssociation VideoInputSource

Select the associated video input source.

Requires user role: `ADMIN`

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

Range: Select one of the video input sources.

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

xConfiguration Audio Input Line [1..2] Channel

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

Requires user role: `ADMIN`

Value space: `<Left/Right/Mono>`

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

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

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

Example: `xConfiguration Audio Input 1 Channel: Left`

xConfiguration Audio Input Line [1..2] Level

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

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

Requires user role: `ADMIN`

Value space: `<0..24>`

Range: Select a value from 0 to 24 dB.

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

xConfiguration Audio Input Line [1..2] LoopSuppression

NOTE: Codec C40/C60 does currently not support Loop Suppression, hence Loop Suppression can be set to Off only.

Requires user role: `ADMIN`

Value space: `<Off>`

Off: Deactivate Loop Suppression.

Example: `xConfiguration Audio Input Line 1 LoopSuppression: Off`

xConfiguration Audio Input Line [1..2] Mode

Set the audio input line mode.

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Enable the Audio Line input.

Off: Disable the Audio Line input.

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

xConfiguration Audio Input Microphone [1..2]/[1..4] EchoControl Mode

NOTE: Codec C40 has two microphone connectors. Codec C60 has four microphone connectors.

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

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

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

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

xConfiguration Audio Input Microphone [1..2]/[1..4] EchoControl NoiseReduction

NOTE: Codec C40 has two microphone connectors. Codec C60 has four microphone connectors.

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

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

Off: Turn off the Noise Reduction.

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

xConfiguration Audio Input Microphone [1..4] EchoControl Dereverberation

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Turn on the dereverberation.

Off: Turn off the dereverberation.

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

xConfiguration Audio Input Microphone [1..2]/[1..4] Equalizer ID

NOTE: Codec C40 has two microphone connectors. Codec C60 has four microphone connectors.

Select the audio input microphone equalizer ID.

Requires user role: `ADMIN`

Value space: `<1..17>`

Range: Select Equalizer ID 1 to 17.

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

xConfiguration Audio Input Microphone [1..2]/[1..4] Equalizer Mode

NOTE: Codec C40 has two microphone connectors. Codec C60 has four microphone connectors.

Set the audio input microphone equalizer mode.

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Enable the equalizer for the audio input microphone.

Off: No equalizer.

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

xConfiguration Audio Input Microphone [1..2]/[1..4] VideoAssociation MuteOnInactiveVideo

NOTE: Codec C40 has two microphone connectors. Codec C60 has four microphone connectors.

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

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

Off: No video source is associated.

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

xConfiguration Audio Input Microphone [1..2]/[1..4] VideoAssociation VideoInputSource

NOTE: Codec C40 has two microphone connectors. Codec C60 has four microphone connectors.

Select the associated video input source.

Requires user role: `ADMIN`

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

Range: Select one of the video input sources.

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

xConfiguration Audio Input Microphone [1..2]/[1..4] Level

NOTE: Codec C40 has two microphone connectors. Codec C60 has four microphone connectors.

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

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

Requires user role: `ADMIN`

Value space: `<0..24>`

Range: Select a value from 0 to 24 dB.

Example: `xConfiguration Audio Input Microphone 1 Level: 15`

xConfiguration Audio Input Microphone [1..2]/[1..4] Mode

NOTE: Codec C40 has two microphone connectors. Codec C60 has four microphone connectors.

Set the audio input microphone mode.

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Enable the microphone connector.

Off: Disable the microphone connector.

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

xConfiguration Audio Input Microphone [1..2]/[1..4] Type

NOTE: Codec C40 has two microphone connectors. Codec C60 has four microphone connectors.

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

Requires user role: `ADMIN`

Value space: `<Microphone/Line>`

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

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

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

xConfiguration Audio Output HDMI [1] Level

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

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

Requires user role: `ADMIN`

Value space: `<-24..0>`

Range: Select a value from -24 to 0dB.

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

xConfiguration Audio Output HDMI [1] Mode

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Enable the audio channel on the HDMI output.

Off: Disable the audio channel on the HDMI output.

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

xConfiguration Audio Output Line [1..2] Channel

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

Requires user role: `ADMIN`

Value space: `<Left/Right/Mono>`

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

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

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

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

xConfiguration Audio Output Line [1..2] Equalizer ID

Select the audio output line equalizer ID.

Requires user role: `ADMIN`

Value space: `<1..8>`

Range: Select EqualizerID 1 to 8.

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

xConfiguration Audio Output Line [1..2] Equalizer Mode

Set the audio output line equalizer mode.

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Enable the equalizer for the audio output line.

Off: No equalizer.

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

xConfiguration Audio Output Line [1..2] Level

Define the output level of the Audio Output Line connector, in steps of 1 dB.

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

Requires user role: `ADMIN`

Value space: `<-24..0>`

Range: Select a value from -24 to 0 dB.

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

xConfiguration Audio Output Line [1..2] Mode

Set the audio output line mode.

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Enable the Audio Line output.

Off: Disable the Audio Line output.

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

xConfiguration Audio Output Line [1] Type

Determine if the Audio Line output will be analog or digital type output. The digital output on the Cisco TelePresence Profile systems are identified as DNAM (Digital Natural Audio Module).

Requires user role: `ADMIN`

Value space: `<Auto/SPDIF>`

Auto: If a Digital NAM is detected then SPDIF mode will be selected, otherwise analog mode will be selected.

SPDIF: Set to SPDIF when you want the line output to be in digital mode.

Example: `xConfiguration Audio Output Line 1 Type: Auto`

xConfiguration Audio Output Line [2] Type

Line output 2 is a dedicated analog output, hence type can be set to analog only.

Requires user role: `ADMIN`

Value space: `<Analog>`

Analog: Can be set to analog only.

Example: `xConfiguration Audio Output Line 2 Type: Analog`

xConfiguration Audio Microphones Mute Enabled

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

Requires user role: `ADMIN`

Value space: `<True/InCallOnly>`

True: Muting of audio is always available.

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

Example: `xConfiguration Audio Microphones Mute Enabled: True`

xConfiguration Audio SoundsAndAlerts KeyTones Mode

The system can produce a sound every time a key on the remote control is pressed.

Requires user role: `USER`

Value space: `<On/Off>`

On: There will be a sound indicator when pressing keys on the remote control.

Off: The remote control Key Tones is switched off.

Example: `xConfiguration Audio SoundsAndAlerts KeyTones Mode: Off`

xConfiguration Audio SoundsAndAlerts RingTone

Select the ring tone for incoming calls.

Requires user role: `USER`

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

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

Example: `xConfiguration Audio SoundsAndAlerts RingTone: Jazz`

xConfiguration Audio SoundsAndAlerts RingVolume

Sets the ring tone volume for an incoming call.

Requires user role: `USER`

Value space: `<0..100>`

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

Example: `xConfiguration Audio SoundsAndAlerts RingVolume: 50`

xConfiguration Audio Volume

Set the volume on the loudspeaker.

Requires user role: `USER`

Value space: `<0..100>`

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

Example: `xConfiguration Audio Volume: 70`

The Cameras configuration

xConfiguration Cameras PowerLine Frequency

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

Requires user role: `ADMIN`

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

Auto: Set to Auto to enable power frequency auto detection in the camera.

50Hz: Set to 50 Hz.

60Hz: Set to 60 Hz.

Example: `xConfiguration Cameras PowerLine Frequency: Auto`

xConfiguration Cameras Camera [1..7] Backlight

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Turn on the camera backlight compensation.

Off: Turn off the camera backlight compensation.

Example: `xConfiguration Cameras Camera 1 Backlight: Off`

xConfiguration Cameras Camera [1..7] Brightness Mode

Set the camera brightness mode.

Requires user role: `ADMIN`

Value space: `<Auto/Manual>`

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

Manual: Enable manual control of the camera brightness, e.g. the level of the brightness level setting will be used for the camera.

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

xConfiguration Cameras Camera [1..7] Brightness Level

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

Requires user role: `ADMIN`

Value space: `<1..31>`

Range: Select a value from 1 to 31.

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

xConfiguration Cameras Camera [1..7] Flip

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

Requires user role: `ADMIN`

Value space: `<Auto/On/Off>`

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

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

Off: Display the video on screen the normal way.

Example: `xConfiguration Cameras Camera 1 Flip: Off`

xConfiguration Cameras Camera [1..7] Focus Mode

Set the camera focus mode.

Requires user role: `ADMIN`

Value space: `<Auto/Manual>`

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

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

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

xConfiguration Cameras Camera [1..7] Gamma Mode

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

Requires user role: `ADMIN`

Value space: `<Auto/Manual>`

Auto: Auto is the default and the recommended setting.

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

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

xConfiguration Cameras Camera [1..7] Gamma Level

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

Requires user role: `ADMIN`

Value space: `<0..7>`

Range: Select a value from 0 to 7.

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

xConfiguration Cameras Camera [1..7] IrSensor

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Enable the IR sensor on the camera.

Off: Disable the IR sensor on the camera.

Example: `xConfiguration Cameras Camera 1 IrSensor: On`

xConfiguration Cameras Camera [1..7] Mirror

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

Requires user role: `ADMIN`

Value space: `<Auto/On/Off>`

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

On: See the selfview in mirror mode, e.g. the selfview is reversed and the experience of selfview is as seeing yourself in a mirror.

Off: See the selfview in normal mode, e.g. the experience of selfview is as seeing yourself as other people see you.

Example: `xConfiguration Cameras Camera 1 Mirror: Off`

xConfiguration Cameras Camera [1..7] Whitebalance Mode

Set the camera whitebalance mode.

Requires user role: `ADMIN`

Value space: `<Auto/Manual>`

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

Manual: Enables manual control of the camera whitebalance, e.g. the level of the whitebalance level setting will be used for the camera.

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

xConfiguration Cameras Camera [1..7] Whitebalance Level

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

Requires user role: `ADMIN`

Value space: `<1..16>`

Range: Select a value from 1 to 16.

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

xConfiguration Cameras Camera [1..7] DHCP

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

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

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

Example: `xConfiguration Cameras Camera 1 DHCP: Off`

The Conference configuration

xConfiguration Conference [1..1] AutoAnswer Mode

Set the AutoAnswer mode.

Requires user role: `ADMIN`

Value space: `<On/Off>`

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

Off: The incoming calls must be answered manually by pressing the OK key or the green Call key on the remote control.

Example: `xConfiguration Conference 1 AutoAnswer Mode: Off`

xConfiguration Conference [1..1] AutoAnswer Mute

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

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

Off: The incoming call will not be muted.

Example: `xConfiguration Conference 1 AutoAnswer Mute: Off`

xConfiguration Conference [1..1] AutoAnswer Delay

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

Requires user role: `ADMIN`

Value space: `<0..50>`

Range: Select a value from 0 to 50 seconds.

Example: `xConfiguration Conference 1 AutoAnswer Delay: 0`

xConfiguration Conference [1..1] MicUnmuteOnDisconnect Mode

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Un-mute the microphones after the call is disconnected.

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

Example: `xConfiguration Conference 1 MicUnmuteOnDisconnect Mode: On`

xConfiguration Conference [1..1] DoNotDisturb Mode

Determine if there should be an alert on incoming calls.

Requires user role: `USER`

Value space: `<On/Off/Timed>`

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

Off: The incoming calls will come through as normal.

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

Example: `xConfiguration DoNotDisturb Mode: Off`

xConfiguration Conference [1..1] FarEndControl Mode

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

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.

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

Example: `xConfiguration Conference 1 FarEndControl Mode: On`

xConfiguration Conference [1..1] FarEndControl SignalCapability

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Enable the far end control signal capability.

Off: Disable the far end control signal capability.

Example: `xConfiguration Conference 1 FarEndControl SignalCapability: On`

xConfiguration Conference [1..1] Encryption Mode

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

Requires user role: `ADMIN`

Value space: `<BestEffort/On/Off>`

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.

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

Off: The system will not use encryption.

Example: `xConfiguration Conference 1 Encryption Mode: BestEffort`

xConfiguration Conference [1..1] DefaultCall Protocol

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

Requires user role: `ADMIN`

Value space: `<H323/Sip>`

H.323: Select H.323 to ensure that calls are set up as H.323 calls.

Sip: Select SIP to ensure that calls are set up as SIP calls.

Example: `xConfiguration Conference 1 DefaultCall Protocol: H323`

xConfiguration Conference [1..1] DefaultCall Rate

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

Requires user role: `ADMIN`

Value space: `<64..6000>`

Range: Select a value between 64 and 6000 kbps

Example: `xConfiguration Conference 1 DefaultCall Rate: 768`

xConfiguration Conference [1..1] MaxTransmitCallRate

Specify the maximum transmit call rate to be used when placing or receiving calls.

Requires user role: `ADMIN`

Value space: `<64..6000>`

Range: Select a value between 64 and 6000 kbps.

Example: `xConfiguration Conference 1 MaxTransmitCallRate: 6000`

xConfiguration Conference [1..1] MaxReceiveCallRate

Specify the maximum receive call rate to be used when placing or receiving calls.

Requires user role: `ADMIN`

Value space: `<64..6000>`

Range: Select a value between 64 and 6000 kbps.

Example: `xConfiguration Conference 1 MaxReceiveCallRate: 6000`

xConfiguration Conference [1..1] VideoBandwidth Mode

Set the conference video bandwidth mode.

Requires user role: `ADMIN`

Value space: `<Dynamic/Static>`

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

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

Example: `xConfiguration Conference 1 VideoBandwidth Mode: Dynamic`

xConfiguration Conference [1..1] VideoBandwidth MainChannel Weight

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

Requires user role: `ADMIN`

Value space: `<1..10>`

Range: 1 to 10.

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

xConfiguration Conference [1..1] VideoBandwidth PresentationChannel Weight

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

Requires user role: `ADMIN`

Value space: `<1..10>`

Range: 1 to 10.

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

xConfiguration Conference [1..1] PacketLossResilience Mode

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Enable the packetloss resilience.

Off: Disable the packetloss resilience.

Example: `xConfiguration Conference 1 PacketLossResilience Mode: On`

xConfiguration Conference [1..1] Presentation Policy

Control how the presentation service is to be performed.

Requires user role: `ADMIN`

Value space: `<LocalRemote/LocalOnly>`

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

LocalOnly: The presentation will only be shown locally.

Example: `xConfiguration Conference 1 Presentation Policy: LocalRemote`

xConfiguration Conference [1..1] Multipoint Mode

Define how the video system handles multipoint video conferences. Basically there are two ways: The video system can use its built-in MultiSite feature (optional), or it can rely on the MultiWay network solution. MultiWay requires that your video network includes an external Multipoint control unit (MCU). The MultiSite feature allows up to four participants (yourself included) plus one additional audio call. An External MCU may let you set up conferences with many participants.

Requires user role: `ADMIN`

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

Off: Multipoint conferences are not allowed.

MultiSite: Use MultiSite for multipoint conferences. If MultiSite is chosen when the MultiSite feature is not available, the Multipoint Mode will be set to Off.

MultiWay: Use MultiWay for multipoint conferences. The Multipoint Mode will be set to Off automatically if the MultiWay service is unavailable, e.g. when a server address is not specified in the NetworkServices MultiWay Address setting.

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

Example: `xConfiguration Conference 1 Multipoint Mode: Auto`

xConfiguration Conference [1..1] IncomingMultisiteCall Mode

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

Requires user role: `ADMIN`

Value space: `<Allow/Deny>`

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

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

Example: `xConfiguration Conference 1 IncomingMultisiteCall Mode: Allow`

The FacilityService configuration

xConfiguration FacilityService Service [1..5] Type

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

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

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

Requires user role: `ADMIN`

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

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

Concierge: Select this option for concierge services.

Helpdesk: Select this option for helpdesk services.

Emergency: Select this option for emergency services.

Security: Select this option for security services.

Catering: Select this option for catering services.

Transportation: Select this option for transportation services.

Example: `xConfiguration FacilityService Service 1 Type: Helpdesk`

xConfiguration FacilityService Service [1..5] Name

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

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

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

Requires user role: `ADMIN`

Value space: `<S: 0, 255>`

Format: String with a maximum of 255 characters.

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

xConfiguration FacilityService Service [1..5] Number

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

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

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

Requires user role: `ADMIN`

Value space: `<S: 0, 255>`

Format: String with a maximum of 255 characters.

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

xConfiguration FacilityService Service [1..5] CallType

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

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

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

Requires user role: `ADMIN`

Value space: `<Video/Audio>`

Video: Select this option for video calls.

Audio: Select this option for audio calls.

Example: `xConfiguration FacilityService Service 1 CallType: Video`

The GPIO configuration

xConfiguration GPIO Pin [1..4] Mode

NOTE: This command is not supported on Codec C40.

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

Requires user role: `ADMIN`

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

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

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

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

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

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

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

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

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

Example: `xConfiguration GPIO Pin 1 Mode: InputNoAction`

The H323 configuration

xConfiguration H323 NAT Mode

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

Requires user role: `ADMIN`

Value space: `<Auto/On/Off>`

Auto: The system will determine if the "NAT Address" or the real IP-address should be used within signalling. This is done to make it possible to place calls to endpoints on the LAN as well as endpoints on the WAN.

On: The system will signal the configured "NAT Address" in place of its own IP-address within Q.931 and H.245. The NAT Server Address will be shown in the startup-menu as: "My IP Address: 10.0.2.1".

Off: The system will signal the real IP Address.

Example: `xConfiguration H323 NAT Mode: Off`

xConfiguration H323 NAT Address

Enter the external/global IP-address to the router with NAT support. Packets sent to the router will then be routed to the system.

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

* Port 1720
* Port 5555-5574
* Port 2326-2485

Requires user role: `ADMIN`

Value space: `<S: 0, 64>`

Format: String with a maximum of 64 characters.

Example: `xConfiguration H323 NAT Address: ""`

xConfiguration H323 Profile [1..1] Authentication Mode

Set the authenticatin mode for the H.323 profile.

Requires user role: `ADMIN`

Value space: `<On/Off>`

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

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

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

xConfiguration H323 Profile [1..1] Authentication LoginName

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

Requires user role: `ADMIN`

Value space: `<S: 0, 50>`

Format: String with a maximum of 50 characters.

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

xConfiguration H323 Profile [1..1] Authentication Password

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

Requires user role: `ADMIN`

Value space: `<S: 0, 50>`

Format: String with a maximum of 50 characters.

Example: `xConfiguration H323 Profile 1 Authentication Password:`

xConfiguration H323 Profile [1..1] CallSetup Mode

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

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

Requires user role: `ADMIN`

Value space: `<Direct/Gatekeeper>`

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

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

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

xConfiguration H323 Profile [1..1] Gatekeeper Discovery

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

Requires user role: `ADMIN`

Value space: `<Manual/Auto>`

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

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

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

xConfiguration H323 Profile [1..1] Gatekeeper Address

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

Requires user role: `ADMIN`

Value space: `<S: 0, 255>`

Format: Only the valid IP address format is accepted. An IP address that contains letters (192.a.2.0) or unvalid IP addresses (192.0.1234.0) will be rejected.

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

xConfiguration H323 Profile [1..1] H323Alias E164

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

Requires user role: `ADMIN`

Value space: `<S: 0, 30>`

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

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

xConfiguration H323 Profile [1..1] H323Alias ID

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

Requires user role: `ADMIN`

Value space: `<S: 0, 49>`

Format: String with a maximum of 49 characters

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

xConfiguration H323 Profile [1..1] PortAllocation

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

Requires user role: `ADMIN`

Value space: `<Dynamic/Static>`

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

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

Example: `xConfiguration H323 Profile 1 PortAllocation: Dynamic`

The Network configuration

xConfiguration Network [1..1] Assignment

Define whether to use DHCP or Static IPv4 assignment.

Requires user role: `ADMIN`

Value space: `<Static/DHCP>`

Static: Set the network assignment to Static and configure the static IPv4 settings (IP Address, SubnetMask and Gateway).

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

Example: `xConfiguration Network 1 Assignment: DHCP`

xConfiguration Network [1..1] DNS Domain Name

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

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

Requires user role: `ADMIN`

Value space: `<S: 0, 64>`

Format: String with a maximum of 64 characters.

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

xConfiguration Network [1..1] DNS Server [1..5] Address

Define the network addresses for DNS servers. Up to 5 addresses may be specified. If the network addresses are unknown, contact your administrator or Internet Service Provider.

Requires user role: `ADMIN`

Value space: `<S: 0, 64>`

Format: String with a maximum of 64 characters.

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

xConfiguration Network [1..1] IPStack

Select which internet protocols the system will support.

Requires user role: `ADMIN`

Value space: `<IPv4/IPv6>`

IPv4: IP version 4 is supported.

IPv6: IP version 6 is supported. The IPv4 settings (IP Address, IP Subnet Mask and Gateway) will be disabled.

Example: `xConfiguration Network 1 IPStack: IPv4`

xConfiguration Network [1..1] IPv4 Address

Enter the static IPv4 network address for the system. Only applicable if the Network Assignment is set to Static.

Requires user role: `ADMIN`

Value space: `<S: 0, 64>`

Format: Only the valid IP address format is accepted. An IP address that contains letters (192.a.2.0) or unvalid IP addresses (192.0.1234.0) will be rejected.

Example: `xConfiguration Network 1 IPv4 Address: "192.0.2.0"`

xConfiguration Network [1..1] IPv4 Gateway

Define the IPv4 network gateway. Only applicable if the Network Assignment is set to Static.

Requires user role: `ADMIN`

Value space: `<S: 0, 64>`

Format: Compact string with a maximum of 64 characters.

Example: `xConfiguration Network 1 IPv4 Gateway: "192.0.2.0"`

xConfiguration Network [1..1] IPv4 SubnetMask

Define the IPv4 network subnet mask. Only applicable if the Network Assignment is set to Static.

Requires user role: `ADMIN`

Value space: `<S: 0, 64>`

Format: Compact string with a maximum of 64 characters.

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

xConfiguration Network [1..1] IPv6 Address

Enter the static IPv6 network address for the system. Only applicable if the Network IPv6 Assignment is set to Static.

Requires user role: `ADMIN`

Value space: `<S: 0, 64>`

Format: The IPv6 address of host name.

Example: `xConfiguration Network 1 IPv6 Address: "ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff"`

xConfiguration Network [1..1] IPv6 Gateway

Define the IPv6 network gateway address. Only applicable if the Network IPv6 Assignment is set to Static.

Requires user role: `ADMIN`

Value space: `<S: 0, 64>`

Format: The IPv6 address of host name.

Example: `xConfiguration Network 1 IPv6 Gateway: "ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff"`

xConfiguration Network [1..1] IPv6 Assignment

Define whether to use Autoconf or Static IPv6 assignment.

Requires user role: `ADMIN`

Value space: `<Static/Autoconf>`

Static: Set the network assignment to Static and configure the static IPv6 settings (IP Address and Gateway).

Autoconf: Enable IPv6 stateless autoconfiguration of the IPv6 network interface. See RFC4862 for a detailed description.

Example: `xConfiguration Network 1 IPv6 Assignment: Autoconf`

xConfiguration Network [1..1] IPv6 DHCPOptions

Retrieves a set of DHCP options from a DHCPv6 server.

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Enable the retrieval of a selected set of DHCP options from a DHCPv6 server.

Off: Set to Off when IPv6 Assignment is set to Static.

Example: `xConfiguration Network 1 IPv6 Gateway: On`

xConfiguration Network [1..1] QoS Mode

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

Requires user role: `ADMIN`

Value space: `<Off/Diffserv>`

Off: No QoS method is used.

Diffserv: When you set the QoS Mode to Diffserv you must configure the Diffserv sub menu settings (Audio, Data, Signalling and Video).

Example: `xConfiguration Network 1 QoS Mode: diffserv`

xConfiguration Network [1..1] QoS Diffserv Audio

The Diffserv Audio defines which priority Audio packets should have in an IP network. Enter a priority, which ranges from 0 to 63 for the packets. The higher the number, the higher the priority. These priorities might be overridden when packets are leaving the network controlled by the local network administrator. NOTE: Requires the Network QoS Mode to be set to Diffserv.

Requires user role: `ADMIN`

Value space: `<0..63>`

Audio: A recommended value is Diffserv Code Point (DSCP) AF41, which equals the value 34. If in doubt, contact your network administrator.

Range: Select a value from 0 to 63.

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

xConfiguration Network [1..1] QoS Diffserv Data

The Diffserv Data defines which priority Data packets should have in an IP network. Enter a priority, which ranges from 0 to 63 for the packets. The higher the number, the higher the priority. These priorities might be overridden when packets are leaving the network controlled by the local network administrator. NOTE: Requires the Network QoS Mode to be set to Diffserv.

Requires user role: `ADMIN`

Value space: `<0..63>`

Data: A recommended value is Diffserv Code Point (DSCP) AF23, which equals the value 22. If in doubt, contact your network administrator.

Range: Select a value from 0 to 63.

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

xConfiguration Network [1..1] QoS Diffserv Signalling

The Diffserv Signalling defines which priority Signalling packets should have in an IP network. Enter a priority, which ranges from 0 to 63 for the packets. The higher the number, the higher the priority. These priorities might be overridden when packets are leaving the network controlled by the local network administrator. NOTE: Requires the Network QoS Mode to be set to Diffserv.

Requires user role: `ADMIN`

Value space: `<0..63>`

Signalling: A recommended value is Diffserv Code Point (DSCP) AF31, which equals the value 26. If in doubt, contact your network administrator.

Range: Select a value from 0 to 63.

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

xConfiguration Network [1..1] QoS Diffserv Video

The Diffserv Video defines which priority Video packets should have in an IP network. Enter a priority, which ranges from 0 to 63 for the packets. The higher the number, the higher the priority. These priorities might be overridden when packets are leaving the network controlled by the local network administrator. NOTE: Requires the Network QoS Mode to be set to Diffserv.

Requires user role: `ADMIN`

Value space: `<0..63>`

Video: A recommended value is Diffserv Code Point (DSCP) AF41, which equals the value 34. If in doubt, contact your network administrator.

Range: Select a value from 0 to 63.

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

xConfiguration Network [1..1] IEEE8021X Mode

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: The 802.1X authentication is enabled.

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

Example: `xConfiguration Network 1 IEEE8021X Mode: Off`

xConfiguration Network [1..1] IEEE8021X TlsVerify

Verification of the server-side certificate of an IEEE802.1x connection against the certificates in the local CA-list when TLS is used. The CA-list must be uploaded to the video system / codec.

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

Requires user role: `ADMIN`

Value space: `<Off/On>`

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

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

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

xConfiguration Network [1..1] IEEE8021X UseClientCertificate

Authentication using a private key/certificate pair during an IEEE802.1x connection. The authentication X.509 certificate must be uploaded to the video system / codec.

Requires user role: `ADMIN`

Value space: `<Off/On>`

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

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

Example: `xConfiguration Network 1 IEEE8021X UseClientCertificate: Off`

xConfiguration Network [1..1] IEEE8021X Identity

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

Requires user role: `ADMIN`

Value space: `<S: 0, 64>`

Format: String with a maximum of 64 characters.

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

xConfiguration Network [1..1] IEEE8021X Password

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

Requires user role: `ADMIN`

Value space: `<S: 0, 32>`

Format: String with a maximum of 32 characters.

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

xConfiguration Network [1..1] IEEE8021X AnonymousIdentity

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

Requires user role: `ADMIN`

Value space: `<S: 0, 64>`

Format: String with a maximum of 64 characters.

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

xConfiguration Network [1..1] IEEE8021X Eap Md5

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

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

Off: The EAP-MD5 protocol is disabled.

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

xConfiguration Network [1..1] IEEE8021X Eap Ttls

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

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

Off: The EAP-TTLS protocol is disabled.

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

xConfiguration Network [1..1] IEEE8021X Eap Tls

Enable or disable the use of EAP-TLS (Transport Layer Security) for IEEE802.1x connections. The EAP-TLS protocol, defined in RFC5216, is considered one of the most secure EAP standards. LAN clients are authenticated using client certificates.

Requires user role: `ADMIN`

Value space: `<Off/On>`

Off: The EAP-TLS protocol is disabled.

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

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

xConfiguration Network [1..1] IEEE8021X Eap Peap

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

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

Off: The EAP-PEAP protocol is disabled.

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

xConfiguration Network [1..1] MTU

Set the Ethernet MTU (Maximum Transmission Unit).

Requires user role: `ADMIN`

Value space: `<576..1500>`

Range: Select a value from 576 to 1500 bytes.

Example: `xConfiguration Network 1 MTU: 1500`

xConfiguration Network [1..1] Speed

Set the Ethernet link speed.

Requires user role: `ADMIN`

Value space: `<Auto/10half/10full/100half/100full/1000full>`

Auto: Autonegotiate link speed.

10half: Force link to 10 Mbps half-duplex.

10full: Force link to 10 Mbps full-duplex.

100half: Force link to 100 Mbps half-duplex.

100full: Force link to 100 Mbps full-duplex.

1000full: Force link to 1 Gbps full-duplex.

Example: `xConfiguration Network 1 Speed: Auto`

xConfiguration Network [1..1] TrafficControl Mode

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Transmit video packets at maximum 20 Mbps. Can be used to smooth out bursts in the outgoing network traffic.

Off: Transmit video packets at link speed.

Example: `xConfiguration Network 1 TrafficControl: On`

xConfiguration Network [1..1] RemoteAccess Allow

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

Requires user role: `ADMIN`

Value space: `<S: 0, 255>`

Format: String with a maximum of 255 characters, comma separated IP adresses or IP range.

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

xConfiguration Network [1..1] VLAN Voice Mode

Set the VLAN voice mode.

Requires user role: `ADMIN`

Value space: `<Auto/Manual/Off>`

Auto: The Cisco Discovery Protocol (CDP), if available, assigns an id to the voice VLAN. If CDP is not available, VLAN is not enabled. The VLAN Voice Mode automatically will be set to Auto when the GUI is used to set the Provisioning Mode to CUCM.

Manual: The VLAN id is set manually using the Network VLAN Voice VlanId setting. If CDP is available, the manually set value will be overruled by the value assigned by CDP.

Off: VLAN is not enabled.

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

xConfiguration Network [1..1] VLAN Voice VlanId

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

Requires user role: `ADMIN`

Value space: `<1..4094>`

Range: Select a value from 1 to 4094.

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

The NetworkPort configuration

xConfiguration NetworkPort [2] Mode

Define if the network port 2 shall be enabled for direct pairing with the Cisco TelePresence Touch for C Series.

Requires user role: `ADMIN`

Value space: `<Inactive/DirectPairing>`

Inactive: Set the NetworkPort 2 to Inactive when no device is connected.

DirectPairing: Set the NetworkPort 2 to DirectPairing when you have a Cisco TelePresence Touch unit connected to the port. This will enable for direct pairing between the touch unit and the codec.

Example: `xConfiguration NetworkPort 2 Mode: Inactive`

The NetworkServices configuration

xConfiguration NetworkServices MultiWay Address

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

Multiway™ can be used in the following situations:

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

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

Requirements: The Codec C20 must be running TC3.0 (or later), Codec C90/C60/C40 must be running TC4.0 (or later), EX90/EX60/MX200 must be running TC4.2 (or later), MX300 must be running TC5.0 (or later),Video Communication Server (VCS) version X5 (or later) and Codian MCU version 3.1 (or later). Endpoints invited to join the Multiway™ conference must support the H.323 routeToMC facility message if in an H.323 call, or SIP REFER message if in a SIP call.

Requires user role: `ADMIN`

Value space: `<S: 0, 255>`

Format: String with a maximum of 255 characters.

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

xConfiguration NetworkServices MultiWay Protocol

Determine the protocol to be used for Multiway calls. NOTE: Requires a restart of the codec.

Requires user role: `ADMIN`

Value space: `<Auto/H323/Sip>`

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

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

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

Example: `xConfiguration NetworkServices MultiWay Protocol: Auto`

xConfiguration NetworkServices H323 Mode

Determine whether the system should be able to place and receive H.323 calls or not. NOTE: Requires a restart of the codec.

Requires user role: `ADMIN`

Value space: `<On/Off>`

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

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

Example: `xConfiguration NetworkServices H323 Mode: On`

xConfiguration NetworkServices HTTP Mode

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: The HTTP protocol is enabled.

Off: The HTTP protocol is disabled.

Example: `xConfiguration NetworkServices HTTP Mode: On`

xConfiguration NetworkServices HTTPS Mode

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: The HTTPS protocol is enabled.

Off: The HTTPS protocol is disabled.

Example: `xConfiguration NetworkServices HTTPS Mode: On`

xConfiguration NetworkServices HTTPS VerifyServerCertificate

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

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.

Off: Do not verify server certificates.

Example: `xConfiguration NetworkServices HTTPS VerifyServerCertificate: Off`

xConfiguration NetworkServices HTTPS VerifyClientCertificate

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

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.

Off: Do not verify client certificates.

Example: `xConfiguration NetworkServices HTTPS VerifyClientCertificate: Off`

xConfiguration NetworkServices HTTPS OCSP Mode

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

Requires user role: `ADMIN`

Value space: `<Off/On>`

Off: Disable OCSP support.

On: Enable OCSP support.

Example: `xConfiguration NetworkServices HTTPS OCSP Mode: Off`

xConfiguration NetworkServices HTTPS OCSP URL

Specify the URL of an OCSP server.

Requires user role: `ADMIN`

Value space: `<S: 0, 255>`

Format: String with a maximum of 255 characters.

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

xConfiguration NetworkServices NTP Mode

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

Requires user role: `ADMIN`

Value space: `<Off/Auto/Manual>`

Off: The system will not use an NTP server.

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

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

Example: `xConfiguration NetworkServices NTP Mode: Manual`

xConfiguration NetworkServices NTP Address

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

Requires user role: `ADMIN`

Value space: `<S: 0, 64>`

Format: String with a maximum of 64 characters.

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

xConfiguration NetworkServices SIP Mode

Determine whether the system should be able to place and receive SIP calls or not. NOTE: Requires a restart of the codec.

Requires user role: `ADMIN`

Value space: `<On/Off>`

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

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

Example: `xConfiguration NetworkServices SIP Mode: On`

xConfiguration NetworkServices SNMP Mode

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

Requires user role: `ADMIN`

Value space: `<Off/ReadOnly/ReadWrite>`

Off: Disable the SNMP network service.

ReadOnly: Enable the SNMP network service for queries only.

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

Example: `xConfiguration NetworkServices SNMP Mode: ReadWrite`

xConfiguration NetworkServices SNMP Host [1..3] Address

Enter the address of up to three SNMP Managers.

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

Requires user role: `ADMIN`

Value space: `<S: 0, 64>`

Format: String with a maximum of 64 characters.

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

xConfiguration NetworkServices SNMP CommunityName

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

Requires user role: `ADMIN`

Value space: `<S: 0, 50>`

Format: String with a maximum of 50 characters.

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

xConfiguration NetworkServices SNMP SystemContact

Enter the name of the Network Services SNMP System Contact.

Requires user role: `ADMIN`

Value space: `<S: 0, 50>`

Format: String with a maximum of 50 characters.

Example: `xConfiguration NetworkServices SNMP SystemContact: ""`

xConfiguration NetworkServices SNMP SystemLocation

Enter the name of the Network Services SNMP System Location.

Requires user role: `ADMIN`

Value space: `<S: 0, 50>`

Format: String with a maximum of 50 characters.

Example: `xConfiguration NetworkServices SNMP SystemLocation: ""`

xConfiguration NetworkServices SSH Mode

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: The SSH protocol is enabled.

Off: The SSH protocol is disabled.

Example: `xConfiguration NetworkServices SSH Mode: On`

xConfiguration NetworkServices SSH AllowPublicKey

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: The SSH public key is allowed.

Off: The SSH public key is not allowed.

Example: `xConfiguration NetworkServices SSH AllowPublicKey: On`

xConfiguration NetworkServices Telnet Mode

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: The Telnet protocol is enabled.

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

Example: `xConfiguration NetworkServices Telnet Mode: Off`

The Phonebook configuration

xConfiguration Phonebook Server [1..1] ID

Enter a name for the external phonebook.

Requires user role: `ADMIN`

Value space: `<S: 0, 64>`

Format: String with a maximum of 64 characters.

Example: `xConfiguration Phonebook Server 1 ID: ""`

xConfiguration Phonebook Server [1..1] Type

Select the phonebook server type.

Requires user role: `ADMIN`

Value space: `<VCS/TMS/Callway/CUCM>`

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

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

Callway: Select Callway if the phonebook is to be provided by the Callway subscription service. Contact your Callway provider for more information.

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

Example: `xConfiguration Phonebook Server 1 Type: TMS`

xConfiguration Phonebook Server [1..1] URL

Enter the address (URL) to the external phonebook server.

Requires user role: `ADMIN`

Value space: `<S: 0, 255>`

Format: String with a maximum of 255 characters.

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

The Provisioning configuration

xConfiguration Provisioning Connectivity

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

Requires user role: `ADMIN`

Value space: `<Internal/External/Auto>`

Internal: Request internal configuration.

External: Request external configuration.

Auto: Automatically discover using NAPTR queries whether internal or external configurations should be requested. If the NAPTR responses have the "e" flag, external configurations will be requested. Otherwise internal configurations will be requested.

Example: `xConfiguration Provisioning Connectivity: Auto`

xConfiguration Provisioning Mode

It is possible to configure the codec (video system) using a provisioning system / an external manager. This allows video conferencing network administrators to manage many video systems simultaneously.

With this setting you choose which type of provisioning system to use. Provisioning can also be switched off. Contact your provisioning system provider/representative for more information.

Requires user role: `ADMIN`

Value space: `<Off/TMS/VCS/CallWay/CUCM/Auto>`

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

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

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

Callway: The video system will be configured using Callway (subscription service).

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

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

Example: `xConfiguration Provisioning Mode: TMS`

xConfiguration Provisioning LoginName

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

Requires user role: `ADMIN`

Value space: `<S: 0, 80>`

Format: String with a maximum of 80 characters.

Example: `xConfiguration Provisioning LoginName: ""`

xConfiguration Provisioning Password

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

Requires user role: `ADMIN`

Value space: `<S: 0, 64>`

Format: String with a maximum of 64 characters.

Example: `xConfiguration Provisioning Password: ""`

xConfiguration Provisioning HttpMethod

Select the HTTP method to be used for the provisioning.

Requires user role: `ADMIN`

Value space: `<GET/POST>`

GET: Select GET when the provisiong server supports GET.

POST: Select POST when the provisiong server supports POST.

Example: `xConfiguration Provisioning HttpMethod: POST`

xConfiguration Provisioning ExternalManager Address

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

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

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

Requires user role: `ADMIN`

Value space: `<S: 0, 64>`

Format: A valid IP address format or DNS name; a compact string with a maximum of 64 characters.

Example: `xConfiguration Provisioning ExternalManager Address: ""`

xConfiguration Provisioning ExternalManager Protocol

Determine whether to use secure management or not.

Requires user role: `ADMIN`

Value space: `<HTTP/HTTPS>`

HTTP: Set to HTTP to disable secure management. Requires HTTP to be enabled in the xConfiguration NetworkServices HTTP Mode setting.

HTTPS: Set to HTTPS to enable secure management. Requires HTTPS to be enabled in the xConfiguration NetworkServices HTTPS Mode setting.

Example: `xConfiguration Provisioning ExternalManager Protocol: HTTP`

xConfiguration Provisioning ExternalManager Path

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

Requires user role: `ADMIN`

Value space: `<S: 0, 255>`

Format: String with a maximum of 255 characters.

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

xConfiguration Provisioning ExternalManager Domain

Enter the SIP domain for the VCS provisioning server.

Requires user role: `ADMIN`

Value space: `<S: 0, 64>`

Format: String with a maximum of 64 characters.

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

The RTP configuration

xConfiguration RTP Ports Range Start

Specify the first port in the range of RTP ports. See also the "H323 Profile [1..1] PortAllocation" command.

Requires user role: `USER`

Value space: `<1024..65502>`

Range: Select a value from 1024 to 65502.

Example: `xConfiguration RTP Ports Range Start: 2326`

xConfiguration RTP Ports Range Stop

Specify the last RTP port in the range. See also the "H323 Profile [1..1] PortAllocation" command.

Requires user role: `USER`

Value space: `<1056..65535>`

Range: Select a value from 1056 to 65535.

Example: `xConfiguration RTP Ports Range Stop: 2486`

The Security configuration

xConfiguration Security Audit Server Address

Enter the external/global IP-address to the audit syslog server. IPv6 is not supported.

NOTE: Requires a restart of the system for any change to take effect.

Requires user role: `AUDIT`

Value space: `<S: 0, 64>`

Format: String with a maximum of 64 characters.

Example: `xConfiguration Security Audit Server Address: ""`

xConfiguration Security Audit Server Port

Enter the port of the syslog server that the system shall send its audit logs to. The default port is 514.

NOTE: Requires a restart of the system for any change to take effect.

Requires user role: `AUDIT`

Value space: `<0..65535>`

Range: Select a value from 0 to 65535.

Example: `xConfiguration Security Audit Server Port: 514`

xConfiguration Security Audit OnError Action

Describes what actions will be taken if connection to the syslog server is lost. This setting is only relevant if Security Audit Logging Mode is set to ExternalSecure.

NOTE: Requires a restart of the system for any change to take effect.

Requires user role: `AUDIT`

Value space: `<Halt/Ignore>`

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

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

Example: `xConfiguration Security Audit OnError Action: Ignore`

xConfiguration Security Audit Logging Mode

Describes where the audit logs are recorded or transmitted.

NOTE: Requires a restart of the system for any change to take effect.

Requires user role: `AUDIT`

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

Off: No audit logging is performed.

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

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

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

Example: `xConfiguration Security Audit Logging Mode: Off`

xConfiguration Security Session ShowLastLogon

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

Requires user role: `ADMIN`

Value space: `<Off/On>`

On: Set to On to enable the possibility to show information about the last session.

Off: Set to Off to disable the possibility to show information about the last session.

Example: `xConfiguration Security Session ShowLastLogon: Off`

xConfiguration Security Session InactivityTimeout

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

Requires user role: `ADMIN`

Value space: `<0..10000>`

Range: Select a value from 0 to 10000 seconds. 0 means that inactivity will not enforce automatically logout.

Example: `xConfiguration Security Session InactivityTimeout: 0`

The SerialPort configuration

xConfiguration SerialPort Mode

Enable/disable the serial port (COM port).

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Enable the serial port.

Off: Disable the serial port.

Example: `xConfiguration SerialPort Mode: On`

xConfiguration SerialPort BaudRate

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

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

Requires user role: `ADMIN`

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

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

Example: `xConfiguration SerialPort BaudRate: 38400`

xConfiguration SerialPort LoginRequired

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

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

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

Example: `xConfiguration SerialPort LoginRequired: On`

The SIP configuration

xConfiguration SIP Profile [1..1] URI

The SIP URI or number is used to address the system. This is the URI that is registered and used by the SIP services to route inbound calls to the system. A Uniform Resource Identifier (URI) is a compact string of characters used to identify or name a resource.

Requires user role: `ADMIN`

Value space: `<S: 0, 255>`

Format: Compact string with a maximum of 255 characters.

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

xConfiguration SIP Profile [1..1] DisplayName

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

Requires user role: `ADMIN`

Value space: `<S: 0, 255>`

Format: String with a maximum of 255 characters.

Example: `xConfiguration SIP Profile 1 DisplayName: ""`

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

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

Requires user role: `ADMIN`

Value space: `<S: 0, 128>`

Format: String with a maximum of 128 characters.

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

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

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

Requires user role: `ADMIN`

Value space: `<S: 0, 128>`

Format: String with a maximum of 128 characters.

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

xConfiguration SIP Profile [1..1] DefaultTransport

Select the transport protocol to be used over the LAN.

Requires user role: `ADMIN`

Value space: `<UDP/TCP/Tls/Auto>`

UDP: The system will always use UDP as the default transport method.

TCP: The system will always use TCP as the default transport method.

Tls: The system will always use TLS as the default transport method. For TLS connections a SIP CA-list can be uploaded to the video system. If no such CA-list is available on the system then anonymous Diffie Hellman will be used.

Auto: The system will try to connect using transport protocols in the following order: TLS, TCP, UDP.

Example: `xConfiguration SIP Profile 1 DefaultTransport: Auto`

xConfiguration SIP Profile [1..1] TlsVerify

For TLS connections a SIP CA-list can be uploaded to the video system.

Requires user role: `ADMIN`

Value space: `<On/Off>`

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

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

Example: `xConfiguration SIP Profile 1 TlsVerify: Off`

xConfiguration SIP Profile [1..1] Outbound

The client initiated connections mechanism for firewall traversal, connection reuse and redundancy. The current version supports http://tools.ietf.org/html/draft-ietf-sip-outbound-20.

Requires user role: `ADMIN`

Value space: `<On/Off>`

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

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

Example: `xConfiguration SIP Profile 1 Outbound: Off`

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

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

Requires user role: `ADMIN`

Value space: `<S: 0, 255>`

Format: Compact string with a maximum of 255 characters. An IP address that contains letters (192.a.2.0) or unvalid IP addresses (192.0.1234.0) will be rejected.

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

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

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

Requires user role: `ADMIN`

Value space: `<Auto/Manual>`

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

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

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

xConfiguration SIP Profile [1..1] Type

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

Requires user role: `ADMIN`

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

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

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

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

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

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

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

Example: `xConfiguration SIP Profile 1 Type: Standard`

The Standby configuration

xConfiguration Standby Control

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

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

Off: The system will not enter standby mode.

Example: `xConfiguration Standby Control: On`

xConfiguration Standby Delay

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

Requires user role: `ADMIN`

Value space: `<1..480>`

Range: Select a value from 1 to 480 minutes.

Example: `xConfiguration Standby Delay: 10`

xConfiguration Standby BootAction

Define the camera position after a restart of the codec.

Requires user role: `ADMIN`

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

None: No action.

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

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

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

Example: `xConfiguration Standby BootAction: DefaultCameraPosition`

xConfiguration Standby StandbyAction

Define the camera position when going into standby mode.

Requires user role: `ADMIN`

Value space: `<None/PrivacyPosition>`

None: No action.

PrivacyPosition: Turns the camera to a sideways position for privacy.

Example: `xConfiguration Standby StandbyAction: PrivacyPosition`

xConfiguration Standby WakeupAction

Define the camera position when leaving standby mode.

Requires user role: `ADMIN`

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

None: No action.

Preset1 to Preset15: When leaving standby the camera position will be set to the position defined by the selected preset.

RestoreCameraPosition: When leaving standby the camera position will be set to the position it had before entering standby.

DefaultCameraPosition: When leaving standby the camera position will be set to the factory default position.

Example: `xConfiguration Standby WakeupAction: RestoreCameraPosition`

The SystemUnit configuration

xConfiguration SystemUnit Name

Enter a System Name to define a name of the system unit. If the H.323 Alias ID is configured on the system then this ID will be used instead of the system name. The system name will be displayed:

1) When the codec is acting as an SNMP Agent.
2) Towards a DHCP server.

Requires user role: `ADMIN`

Value space: `<S: 0, 50>`

Format: String with a maximum of 50 characters.

Example: `xConfiguration SystemUnit Name: "Meeting Room"`

xConfiguration SystemUnit MenuLanguage

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

Requires user role: `USER`

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

Example: `xConfiguration SystemUnit MenuLanguage: English`

xConfiguration SystemUnit ContactInfo Type

Describes which parameter to put in the status field in the upper left corner on the screen display. The information can also be read with the command xStatus SystemUnit ContactInfo.

Requires user role: `ADMIN`

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

Auto: Shows the address which another system can dial to reach this system, depending on the default call protocol and system registration.

None: Do not show any contact information.

IPv4: Shows the IPv4 address as the contact information.

IPv6: Shows the IPv6 address as the contact information.

H323Id: Shows the H323 ID as the contact information.

E164Alias: Shows the H323 E164 Alias as the contact information.

SipUri: Shows the SIP URI as the contact information.

SystemName: Shows the system name as the contact information.

Example: `xConfiguration SystemUnit ContactInfo Type: Auto`

xConfiguration SystemUnit Type

Select whether the video system is for personal use or to be used in a multiuser environment. It is highly recommended to use the default setting.

Requires user role: `ADMIN`

Value space: `<Personal/Shared>`

Personal: Set to Personal when the system is for personal use.

Shared: Set to Shared when the system is used in a multiuser environment.

Example: `xConfiguration SystemUnit Type: Shared`

xConfiguration SystemUnit CallLogging Mode

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Enable logging.

Off: Disable logging.

Example: `xConfiguration SystemUnit CallLogging Mode: On`

xConfiguration SystemUnit IrSensor

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

Requires user role: `ADMIN`

Value space: `<On/Off/Auto>`

On: Enable the IR sensor on the codec.

Off: Disable the IR sensor on the codec.

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

Example: `xConfiguration SystemUnit IrSensor: Auto`

The Time configuration

xConfiguration Time Zone

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

Requires user role: `USER`

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

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

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

xConfiguration Time TimeFormat

Set the time format.

Requires user role: `USER`

Value space: `<24H/12H>`

24H: Set the time format to 24 hours.

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

Example: `xConfiguration Time TimeFormat: 24H`

xConfiguration Time DateFormat

Set the date format.

Requires user role: `USER`

Value space: `<DD_MM_YY/MM_DD_YY/YY_MM_DD>`

DD_MM_YY: The date January 30th 2010 will be displayed: 30.01.10

MM_DD_YY: The date January 30th 2010 will be displayed: 01.30.10

YY_MM_DD: The date January 30th 2010 will be displayed: 10.01.30

Example: `xConfiguration Time DateFormat: DD_MM_YY`

The UserInterface configuration

xConfiguration UserInterface TouchPanel DefaultPanel

Select whether to display the list of contacts or the list of scheduled meetings on the Touch panel as default.

Requires user role: `USER`

Value space: `<ContactList/MeetingList>`

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

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

Example: `xConfiguration UserInterface TouchPanel DefaultPanel: ContactList`

The Video configuration

xConfiguration Video Input Source [1..3] Name

Enter a name for the video input source.

Requires user role: `ADMIN`

Value space: `<S: 0, 50>`

Format: String with a maximum of 50 characters.

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

xConfiguration Video Input Source [1] Connector

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

Requires user role: `ADMIN`

Value space: `<HDMI>`

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

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

xConfiguration Video Input Source [2] Connector

NOTE: Codec C40 has one DVI input (DVI-I 3). Codec C60 has two DVI inputs (DVI-I 2 and 3).

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

Requires user role: `ADMIN`

Value space: `<HDMI/DVI>`

HDMI: Select HDMI when you want to use the HDMI 2 as input source 2.

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

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

xConfiguration Video Input Source [3] Connector

NOTE: Codec C40 has one DVI input (DVI-I 3). Codec C60 has two DVI inputs (DVI-I 2 and 3).

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

Requires user role: `ADMIN`

Value space: `<DVI/Composite/YC>`

DVI: Select DVI-I when you want to use the DVI-I 3 as input source 3.

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

YC: Select YC when you want to use the S-Video (YC) as input source 3. Connect to the two connectors marked Y/Comp and C.

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

xConfiguration Video Input Source [1..3] Type

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

Requires user role: `ADMIN`

Value space: `<other/camera/PC/DVD/document _ camera>`

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

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

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

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

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

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

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

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Enable camera control.

Off: Disable camera control.

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

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

Select the ID of the camera in the Visca chain that is connected to this camera source. The CameraId setting represents the camera's position in the Visca chain.

Requires user role: `ADMIN`

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

Range: Select the ID of the camera in the Visca chain.

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

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

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

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

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

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

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

Typical resolutions used for different optimal definition profiles, call rates and frame rates

Frame rate	Optimal Definition Profile	Call rate						
		256 kbps	*768 kbps*	*1152 kbps*	*1472 kbps*	*2560 kbps*	*4 Mbps*	*6 Mbps*
30 fps	*Normal*	512×288	1024×576	1280×720	1280×720	1920×1080	1920×1080	1920×1080
	Medium	640×368	1280×720	1280×720	1280×720	1920×1080	1920×1080	1920×1080
	High	768×448	1280×720	1280×720	1920×1080	1920×1080	1920×1080	1920×1080
60 fps	*Normal*	256×144	512×288	768×448	1024×576	1280×720	1280×720	1280×720
	Medium	256×144	768×448	1024×576	1024×576	1280×720	1280×720	1280×720
	High	512×288	1024×576	1280×720	1280×720	1280×720	1280×720	1280×720

Requires user role: `ADMIN`

Value space: `<Normal/Medium/High>`

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

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

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

Example: `xConfiguration Video Input Source 1 OptimalDefinition Profile: Normal`

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

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

Requires user role: `ADMIN`

Value space: `<512 _ 288/768 _ 448/1024 _ 576/1280 _ 720/Never>`

512_288: Set the threshold to 512x288.

768_448: Set the threshold to 768x448.

1024_576: Set the threshold to 1024x576.

1280_720: Set the threshold to 1280x720.

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

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

xConfiguration Video Input Source [1..3] Quality

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

Requires user role: `ADMIN`

Value space: `<Motion/Sharpness>`

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

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

Example: `xConfiguration Video Input Source 1 Quality: Motion`

xConfiguration Video DefaultPresentationSource

Define which video input source shall be used as the default presentation source (when you press the Presentation key on the remote control). The input source is configured to a video input connector.

Requires user role: `USER`

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

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

Example: `xConfiguration Video DefaultPresentationSource: 3`

xConfiguration Video Input DVI [3]/[2..3] Type

NOTE: Codec C40 has one DVI input (DVI-I 3). Codec C60 has two DVI inputs (DVI-I 2 and 3).

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

This setting should also be used if the video input is an analog component (YPbPr) type signal. This is used by some cameras (Sony EVI-HD1) and DVD/Blu-ray players. Since it is not possible to auto detect the difference between AnalogRGB and AnalogYPbPr, the AnalogYPbPr setting must be selected.

Requires user role: `ADMIN`

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

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

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

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

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

Example: `xConfiguration Video Input DVI 3 Type: AutoDetect`

xConfiguration Video Layout Scaling

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

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Let the system automatically adjust aspect ratio.

Off: No adjustment of the aspect ratio.

Example: `xConfiguration Video Layout Scaling: On`

xConfiguration Video Layout ScaleToFrame

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

Requires user role: `ADMIN`

Value space: `<Manual/MaintainAspectRatio/StretchToFit>`

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

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

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

Example: `xConfiguration Video Layout ScaleToFrame: MaintainAspectRatio`

xConfiguration Video Layout ScaleToFrameThreshold

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

Requires user role: `ADMIN`

Value space: `<0..100>`

Range: Select a value from 0 to 100 percent.

Example: `xConfiguration Video Layout ScaleToFrameThreshold: 5`

xConfiguration Video SelfviewPosition

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

Requires user role: `ADMIN`

Value space: `<UpperLeft/UpperRight/LowerLeft/LowerRight/CenterRight>`

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

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

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

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

CenterRight: The selfview PiP will appear in to the right side of the screen, in center.

Example: `xConfiguration Video SelfviewPosition: LowerRight`

xConfiguration Video Layout LocalLayoutFamily

Select which video layout family to be used locally.

Requires user role: `ADMIN`

Value space: `<Auto/FullScreen/Equal/PresentationSmallSpeaker/PresentationLargeSpeaker>`

Auto: The default layout family, as given by the layout database, will be used as the local layout. For more information about the layout database, see the xCommand Video Layout LoadDb command.

FullScreen: The FullScreen layout family will be used as the local layout.

Equal: The Equal layout family will be used as the local layout.

PresentationSmallSpeaker: The PresentationSmallSpeaker layout family will be used as the local layout.

PresentationLargeSpeaker: The PresentationLargeSpeaker layout family will be used as the local layout.

Example: `xConfiguration Video Layout LocalLayoutFamily: Auto`

xConfiguration Video Layout RemoteLayoutFamily

Select which video layout family to be used for the remote participants.

Requires user role: `ADMIN`

Value space: `<Auto/FullScreen/Equal/PresentationSmallSpeaker/PresentationLargeSpeaker>`

Auto: The default layout family, as given by the local layout database, will be used as the remote layout. For more information about the layout database, see the command: xCommand Video Layout LoadDb.

FullScreen: The FullScreen layout family will be used as the remote layout.

Equal: The Equal layout family will be used as the remote layout.

PresentationSmallSpeaker: The PresentationSmallSpeaker layout family will be used as the remote layout.

PresentationLargeSpeaker: The PresentationLargeSpeaker layout family will be used as the remote layout.

Example: `xConfiguration Video Layout RemoteLayoutFamily: Auto`

xConfiguration Video MainVideoSource

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

Requires user role: `USER`

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

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

Example: `xConfiguration Video MainVideoSource: 1`

xConfiguration Video Monitors

Set the monitor layout mode.

Requires user role: `ADMIN`

Value space: `<Single/Dual/DualPresentationOnly>`

Single: The same layout is shown on all monitors.

Dual: The layout is distributed on two monitors.

DualPresentationOnly: All participants in the call will be shown on the first monitor, while the presentation (if any) will be shown on the second monitor.

Example: `xConfiguration Video Monitors: Single`

xConfiguration Video OSD Mode

The Video OSD (On Screen Display) Mode lets you define if information and icons should be displayed on screen.

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Display the on screen menus, icons and indicators.

Off: Hide the on screen menus, icons and indicators.

Example: `xConfiguration Video OSD Mode: On`

xConfiguration Video OSD AutoSelectPresentationSource

Determine if the presentation source should be automatically selected.

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Enable automatic selection of the presentation source.

Off: Disable automatic selection of the presentation source.

Example: `xConfiguration Video OSD AutoSelectPresentationSource: Off`

xConfiguration Video OSD TodaysBookings

This setting can be used to display the systems bookings for today on the main OSD menu. This requires that the system is bookable by an external booking system, like Cisco TelePresence Management Suite (TMS).

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Displays information about this systems bookings on screen.

Off: Do not display todays bookings.

Example: `xConfiguration Video OSD TodaysBookings: Off`

xConfiguration Video OSD MyContactsExpanded

Set how the local contacts will be displayed in the phone book dialog in the OSD (On Screen Display).

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: The local contacts in the phone book will be shown in the top level of the phonebook dialog.

Off: The local contacts will be placed in a separate folder called MyContacts in the phonebook dialog.

Example: `xConfiguration Video OSD MyContactsExpanded: Off`

xConfiguration Video OSD Output

The Video OSD (On Screen Display) Output lets you define which monitor should display the on screen menus, information and icons. By default the OSD is sent to the monitor connected to the Video OSD Output 1. If you cannot see the OSD on screen, then you must re-configure the OSD Output. You can do this by entering a key sequence on the remote control, from the web interface, or by a command line interface.

Using the remote control: Press the Disconnect key followed by: * # * # 0 x # (where x is output 1 to 2).

Using the web interface: Open a web browser and enter the IP address of the codec. Open the Advanced Configuration menu and navigate to Video OSD Output and select the video output.

Using a command line interface: Open a command line interface and connect to the codec (if in doubt of how to do this, see the API Guide for the codec). Enter the command: xConfiguration Video OSD Output [1..2] (select the OSD Output)

Requires user role: `ADMIN`

Value space: `<1/2>`

Range: Select 1 for HDMI 1 output, or select 2 for DVI-I 2 output.

Example: `xConfiguration Video OSD Output: 1`

xConfiguration Video OSD InputMethod InputLanguage

The codec can be enabled for Cyrillic input characters in the menus on screen. NOTE: Requires that xConfiguration Video OSD inputMethod Cyrillic is set to On.

Requires user role: `ADMIN`

Value space: `<Latin/Cyrillic>`

Latin: Latin characters can be entered when using the remote control (default).

Cyrillic: Cyrillic characters can be entered using the remote control. NOTE: Requires a Cisco TelePresence Remote Control with Cyrillic fonts.

Example: `xConfiguration Video OSD InputMethod InputLanguage: Latin`

xConfiguration Video OSD InputMethod Cyrillic

Set the Cyrillic mode for the menu input language in the menus on screen.

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Cyrillic mode is available as a menu input language in the menus on screen. This will enable the setting xConfiguration Video OSD InputMethod InputLanguage.

Off: Cyrillic mode is NOT available as a menu input language in the menus on screen.

Example: `xConfiguration Video OSD InputMethod Cyrillic: Off`

xConfiguration Video OSD LoginRequired

Determine if the system should require the user to login before accessing the On Screen Display (OSD). If enabled, the user must enter his username and his PIN. After the user has logged in he can only execute to the configurations changes and commands allowed by his Role.

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: The user must log in to access the On Screen Display (OSD).

Off: No login to the OSD is required.

Example: `xConfiguration Video OSD LoginRequired: Off`

xConfiguration Video AllowWebSnapshots

Allow or disallow snapshots being taken of the local input sources, remote sites and presentation channel. If allowed, the web interface Call Control page will show snapshots both when idle and in a call.

NOTE: This feature is disabled by default, and must be enabled from the On Screen Display (OSD), from a directly connected Touch controller, or via the codec's serial port (COM).

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Web snapshots can be captured and displayed on the web interface.

Off: Capturing web snapshots is not allowed.

Example: `xConfiguration Video AllowWebSnapshots: Off`

xConfiguration Video Output HDMI [1] CEC Mode

The HDMI outputs support Consumer Electronics Control (CEC). When set to on (default is off), and the monitor connected to the HDMI output is CEC compatible and CEC is configured, the system will use CEC to set the monitor in standby when the system enters standby. Likewise the system will wake up the monitor when the system wakes up from standby. Please note that the different manufacturers uses different marketing names for CEC: Anynet+ (Samsung); Aquos Link (Sharp); BRAVIA Sync (Sony); HDMI-CEC (Hitachi); Kuro Link (Pioneer); CE-Link and Regza Link (Toshiba); RIHD (Onkyo); SimpLink (LG); HDAVI Control, EZ-Sync, VIERA Link (Panasonic); EasyLink (Philips); and NetCommand for HDMI (Mitsubishi).

Requires user role: `ADMIN`

Value space: `<On/Off>`

On: Enable CEC control.

Off: Disable CEC control.

Example: `xConfiguration Video Output HDMI 1 CEC Mode: Off`

xConfiguration Video Output HDMI [1] MonitorRole

The HDMI monitor role describes what video stream will be shown on the monitor connected to the video output HDMI connector. Applicable only if the "Video > Monitors" configuration is set to dual.

Requires user role: `ADMIN`

Value space: `<First/Second/PresentationOnly>`

First: Show main video stream.

Second: Show presentation video stream if active, or other participants.

PresentationOnly: Show presentation video stream if active, and nothing else.

Example: `xConfiguration Video Output HDMI 1 MonitorRole: First`

xConfiguration Video Output HDMI [1] OverscanLevel

Some TVs or other monitors may not display the whole image sent out on the systems video output, but cuts the outer parts of the image. In this case this setting can be used to let the system not use the outer parts of video resolution. Both the video and the OSD menu will be scaled in this case.

Requires user role: `ADMIN`

Value space: `<Medium/High/None>`

Medium: The system will not use the outer 3% of the output resolution.

High: The system will not use the outer 6% of the output resolution

None: The system will use all of the output resolution.

Example: `xConfiguration Video Output HDMI 1 OverscanLevel: None`

xConfiguration Video Output HDMI [1] Resolution

Select the preferred resolution for the monitor connected to the video output HDMI connector. This will force the resolution on the monitor.

Requires user role: `ADMIN`

Value space: `<Auto/640_480_60/800_600_60/1024_768_60/1280_1024_60/1280_720_50/1280_720_60/1920_1080_50/1920_1080_60/1280_768_60/1360_768_60/1366_768_60/1600_1200_60/1680_1050_60/1920_1200_60>`

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

Range: 640x480@60p, 800x600@60p, 1024x768@60p, 1280x1024@60p, 1280x720@50p, 1280x720@60p, 1920x1080@50p, 1920x1080@60p, 1280x768@60p, 1360x768@60p, 1366x768@60p, 1600x1200@60p, 1680x1050@60p, 1920x1200@60p

Example: `xConfiguration Video Output HDMI 1 Resolution: 1920_1080_60`

xConfiguration Video Output DVI [2] MonitorRole

The DVI monitor role describes what video stream will be shown on the monitor connected to the video output DVI-I connector. Applicable only if the "Video > Monitors" configuration is set to dual.

Requires user role: `ADMIN`

Value space: `<First/Second/PresentationOnly>`

First: Show main video stream.

Second: Show presentation video stream if active, or other participants.

PresentationOnly: Show presentation video stream if active, and nothing else.

Example: `xConfiguration Video Output DVI 2 MonitorRole: Second`

xConfiguration Video Output DVI [2] OverscanLevel

Some TVs or other monitors may not display the whole image sent out on the systems video output, but cuts the outer parts of the image. In this case this setting can be used to let the system not use the outer parts of video resolution. Both the video and the OSD menu will be scaled in this case.

Requires user role: `ADMIN`

Value space: `<Medium/High/None>`

Medium: The system will not use the outer 3% of the output resolution.

High: The system will not use the outer 6% of the output resolution

None: The system will use all of the output resolution.

Example: `xConfiguration Video Output DVI 2 OverscanLevel: None`

xConfiguration Video Output DVI [2] Resolution

Select the preferred resolution for the monitor connected to the video output DVI-I connector. This will force the resolution on the monitor.

Requires user role: `ADMIN`

Value space: `<Auto/640_480_60/800_600_60/1024_768_60/1280_1024_60/1280_720_50/1280_720_60/1920_1080_50/1920_1080_60/1280_768_60/1360_768_60/1366_768_60/1600_1200_60/1680_1050_60/1920_1200_60>`

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

Range: 640x480@60p, 800x600@60p, 1024x768@60p, 1280x1024@60p, 1280x720@50p, 1280x720@60p, 1920x1080@50p, 1920x1080@60p, 1280x768@60p, 1360x768@60p, 1366x768@60p, 1600x1200@60p, 1680x1050@60p, 1920x1200@60p

Example: `xConfiguration Video Output DVI 2 Resolution: 1024_768_60`

xConfiguration Video Output Composite [3] MonitorRole

NOTE: This command is not supported on Codec C40.

The Composite monitor role describes what video stream will be shown on the monitor connected to the video output Composite connector. Applicable only if the monitor configuration is set to dual.

Requires user role: `ADMIN`

Value space: `<First/Second/PresentationOnly>`

First: Show main video stream.

Second: Show presentation video stream if active, or other participants.

PresentationOnly: Show presentation video stream if active, and nothing else.

Example: `xConfiguration Video Output Composite 3 MonitorRole: First`

xConfiguration Video Output Composite [3] OverscanLevel

NOTE: This command is not supported on Codec C40.

Some TVs or other monitors may not display the whole image sent out on the systems video output, but cuts the outer parts of the image. In this case this setting can be used to let the system not use the outer parts of video resolution. Both the video and the OSD menu will be scaled in this case.

Requires user role: `ADMIN`

Value space: `<Medium/High/None>`

Medium: The system will not use the outer 3% of the output resolution.

High: The system will not use the outer 6% of the output resolution

None: The system will use all of the output resolution.

Example: `xConfiguration Video Output Composite 3 OverscanLevel: None`

xConfiguration Video Output Composite [3] Resolution

NOTE: This command is not supported on Codec C40.

Select the preferred resolution for the monitor connected to the video output Composite connector. This will force the resolution on the monitor.

Requires user role: `ADMIN`

Value space: `<PAL/NTSC>`

Range: PAL, NTSC

Example: `xConfiguration Video Output Composite 3 Resolution: NTSC`

xConfiguration Video Selfview

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

Requires user role: `USER`

Value space: `<On/Off>`

On: Display selfview on screen.

Off: Do not display selfview on screen.

Example: `xConfiguration Video Selfview: On`

xConfiguration Video WallPaper

Select a background image for the video screen when idle. The background image on the Touch controller is not changed.

Requires user role: `USER`

Value space: `<None/Growing/Summersky/Custom/Waves>`

None: There will not be a background image on the screen.

Summersky, Growing, Waves: The selected background image will be shown on the screen.

Custom: If a custom wallpaper is uploaded to the codec, it will be used as background image on the screen. If not, there will be no background image.

Use the web interface to upload a custom wallpaper to the codec.

1) On the codec: With a remote control, open the menu on screen and go to Home > Settings > System information to find the IP address. With a Touch controller, tap More > Settings > System Information to find the IP address.

2) On your computer: Open a web browser and enter the IP address of the codec in the address bar. Hover the mouse over the Configuration tab and select "Wallpaper". Browse for the file and press the "Upload" button. The maximum supported resolution is 1920x1200.

Example: `xConfiguration Video Wallpaper: Summersky`

The Experimental configuration

The Experimental settings are for testing only and should not be used unless agreed with Cisco. These settings are not documented and WILL change in later releases.

Chapter 4

Description of the xCommand commands

Description of the xCommands commands

In the following pages you will 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/telepresence/docs

The Audio commands

xCommand Audio Equalizer List

Shows the current equalizer settings for the codec.

Requires user role: `USER`

Parameters:

EqualizerId: <1..8>

Example:

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

xCommand Audio Equalizer Update

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

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

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

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

Requires user role: `USER`

Parameters:

EqualizerId(r): <1..8>

Section(r): <1..6>

FilterType(r): <highpass/highshelf/lowpass/lowshelf/none/peaking>

Frequency(r): <S: 0, 32>

Q(r): <S: 0, 32>

Gain(r): <S: 0, 32>

Example:

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

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

xCommand Audio Microphones Mute

Mute all microphones.

Requires user role: `USER`

Example:

```
xCommand Audio Microphones Mute
*r AudioMicrophonesMuteResult (status=OK):
** end
```

xCommand Audio Microphones Unmute

Unmute microphones.

Requires user role: `USER`

Example:

```
xCommand Audio Microphones Unmute
*r AudioMicrophonesUnmuteResult (status=OK):
** end
```

xCommand Audio LocalInput Add

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

InputId: A unique identifier for the local input.

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

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

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

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

AGC: Automatic Gain Control.

Mute: Mutes the mix of input connectors.

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

Requires user role: `USER`

Parameters:

InputId: <0..65534>

Name: <S: 0, 255>

MixerMode: <Auto/Fixed/GainShared>

AGC: <On/Off>

Mute: <On/Off>

Channels: <1..2>

Example:

```
xCommand Audio LocalInput Add
OK
*r AudioInputGroupAddResult (status=OK):
 InputId: 2
** end
```

xCommand Audio LocalInput Update

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

InputId: A unique identifier for the local input.

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

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

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

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

AGC: Automatic Gain Control.

Mute: Mutes the mix of input connectors.

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

Requires user role: `USER`

Parameters:

InputId(r): <0..65534>

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

MixerMode(r): <Auto/Fixed/GainShared>

AGC(r): <On/Off>

Mute(r): <On/Off>

Channels(r): <1..2>

Example:

```
xCommand Audio LocalInput Update InputId: 2 Name: "Microphone" MixerMode:
GainShared AGC: Off Mute: Off Channels: 1
OK
*r AudioInputGroupUpdateResult (status=OK):
** end
```

xCommand Audio LocalInput Remove

Remove the local input given by the input ID.

InputId: A unique identifier for the local input.

Requires user role: `USER`

Parameters:

InputId(r): <0..65534>

Example:

```
xCommand Audio LocalInput Remove InputID: 2
OK
*r AudioInputGroupRemoveResult (status=OK):
** end
```

xCommand Audio LocalInput AddConnector

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

InputId: A unique identifier for the local input.

ConnectorType: Select the connector type.

ConnectorId: Select a connector.

Requires user role: `USER`

Parameters:

InputId(r): <0..65534>

ConnectorType(r): <HDMI/Line/Microphone>

ConnectorId(r): <1..8>

Example:

```
xCommand Audio LocalInput AddConnector InputId: 3 ConnectorType: Line
ConnectorId: 1
OK
*r AudioInputGroupAddConnectorResult (status=OK):
** end
```

xCommand Audio LocalInput RemoveConnector

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

InputId: A unique identifier for the local input.

ConnectorType: Select the connector type.

ConnectorId: Select the connector.

Requires user role: `USER`

Parameters:

InputId(r): <0..65534>

ConnectorType(r): <HDMI/Line/Microphone>

ConnectorId(r): <1..8>

Example:

```
xCommand Audio LocalInput RemoveConnector InputId: 3 ConnectorType: Line
ConnectorId: 1
OK
*r AudioInputGroupRemoveConnectorResult (status=OK):
** end
```

xCommand Audio LocalOutput Add

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

OutputId: A unique identifier for the local output.

Name: Choose a name that describes the local output.

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

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

Requires user role: `USER`

Parameters:

OutputId: <0..65534>

Name: <S: 0, 255>

Loudspeaker: <On/Off>

Channels: <1..2>

Example:

```
xCommand Audio LocalOutput Add
OK
*r AudioOutputGroupAddResult (status=OK):
 OutputId: 47
** end
```

xCommand Audio LocalOutput Update

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

OutputId: A unique identifier for the local output.

Name: Choose a name that describes the local output.

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

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

Requires user role: `USER`

Parameters:

OutputId(r): <0..65534>

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

Loudspeaker(r): <On/Off>

Channels(r): <1..2>

Example:

```
xCommand Audio LocalOutput Update OutputId: 5 Name: "Loudspeaker"
Loudspeaker: On Channels: 2
OK
*r AudioOutputGroupUpdateResult (status=OK):
** end
```

xCommand Audio LocalOutput Remove

Remove the local output given by the output ID.

OutputId: A unique identifier for the local output.

Requires user role: `USER`

Parameters:

OutputId(r): <0..65534>

Example:

```
xCommand Audio LocalOutput Remove OutputId: 6
OK
*r AudioOutputGroupRemoveResult (status=OK):
** end
```

xCommand Audio LocalOutput AddConnector

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

OutputId: A unique identifier for the local output.

ConnectorType: Select the connector type.

ConnectorId: Select a connector.

Requires user role: `USER`

Parameters:

OutputId(r): <0..65534>

ConnectorType(r): <HDMI/Line>

ConnectorId(r): <1..8>

Example:

```
xCommand Audio LocalOutput AddConnector OutputId:5 ConnectorType: Line
ConnectorId:1
OK
*r AudioOutputGroupAddConnectorResult (status=OK):
** end
```

xCommand Audio LocalOutput RemoveConnector

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

OutputId: A unique identifier for the local output.

ConnectorType: Select the connector type.

ConnectorId: Select the connector.

Requires user role: `USER`

Parameters:

OutputId(r): <0..65534>

ConnectorType(r): <HDMI/Line>

ConnectorId(r): <1..8>

Example:

```
xCommand Audio LocalOutput RemoveConnector OutputId:5 ConnectorType: Line
ConnectorId:1
OK
*r AudioOutputGroupRemoveConnectorResult (status=OK):
** end
```

xCommand Audio LocalOutput ConnectInput

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

OutputId: A unique identifier for the local output.

InputId: A unique identifier for the local input.

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

Requires user role: `USER`

Parameters:

OutputId(r): <0..65534>

InputId(r): <0..65534>

InputGain: <-54..15>

Example:

```
xCommand Audio LocalOutput ConnectInput OutputId:6 InputId:3
OK
*r AudioOutputGroupConnectInputResult (status=OK):
** end
```

xCommand Audio LocalOutput UpdateInputGain

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

OutputId: A unique identifier for the local output.

InputId: A unique identifier for the local input.

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

Requires user role: `ADMIN`

Parameters:

OutputId(r): <0..65534>

InputId(r): <0..65534>

InputGain: <-54..15>

Example:

```
xCommand Audio LocalOutput UpdateInputGain OutputId:6 InputId:3
OK
*r AudioOutputGroupUpdateInputGainResult (status=OK):
** end
```

xCommand Audio LocalOutput DisconnectInput

Disconnect a local or remote input from a local output.

OutputId: A unique identifier for the local output.

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

Requires user role: `USER`

Parameters:

OutputId(r): <0..65534>

InputId(r): <0..65534>

Example:

```
xCommand Audio LocalOutput DisconnectInput OutputId:6 InputId:3
OK
*r AudioOutputGroupDisconnectInputResult (status=OK):
** end
```

xCommand Audio RemoteOutput ConnectInput

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

OutputId: A unique identifier for the local output.

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

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

Requires user role: `USER`

Parameters:

OutputId(r): <0..65534>

InputId(r): <0..65534>

InputGain: <-54..15>

Example:

```
xCommand Audio RemoteOutput ConnectInput OutputId:10 InputId:8
OK
*r AudioRemoteOutputGroupConnectInputResult (status=OK):
** end
```

xCommand Audio RemoteOutput UpdateInputGain

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

OutputId: A unique identifier for the local output.

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

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

Requires user role: `ADMIN`

Parameters:

OutputId(r): <0..65534>

InputId(r): <0..65534>

InputGain: <-54..15>

Example:

```
xCommand Audio RemoteOutput UpdateInputGain OutputId:6 InputId:3
OK
*r AudioRemoteOutputGroupUpdateInputGainResult (status=OK):
** end
```

xCommand Audio RemoteOutput DisconnectInput

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

OutputId: A unique identifier for the local output.

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

Requires user role: `USER`

Parameters:

OutputId(r): <0..65534>

InputId(r): <0..65534>

Example:

```
xCommand Audio RemoteOutput DisconnectInput OutputId:10 InputId:8
OK
*r AudioRemoteOutputGroupDisconnectInputResult (status=OK):
** end
```

xCommand Audio Setup Clear

Remove all local inputs and local outputs.

Requires user role: `USER`

Example:

```
xCommand Audio Setup Clear
*r AudioSetupClearResult (status=OK):
** end
```

xCommand Audio Sound Play

Play the specified audio sound.

Requires user role: `USER`

Parameters:

Sound(r): <Busy/CallWaiting/Dial/KeyTone/Ringing/SpecialInfo/TelephoneCall/VideoCall>
Loop: <On/Off>

Example:

```
xCommand Audio Sound Play Sound: Ringing
*r AudioSoundPlayResult (status=OK):
** end
```

xCommand Audio Sound Stop

Stop playing audio sound.

Requires user role: `USER`

Example:

```
xCommand Audio Sound Stop
*r AudioSoundStopResult (status=OK):
** end
```

xCommand Audio Vumeter Start

Start collecting VU meter information for connector given by type and ID.

Requires user role: `USER`

Parameters:

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

Example:

```
xCommand Audio Vumeter Start ConnectorType: Microphone ConnectorId: 1
*r AudioVumeterStartResult (status=OK):
** end
```

xCommand Audio Vumeter Stop

Stop collecting VU meter information for connector given by type and ID.

Requires user role: `USER`

Parameters:

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

Example:

```
xCommand Audio Vumeter Stop ConnectorType: Microphone ConnectorId: 1
*r AudioVumeterStopResult (status=OK):
** end
```

xCommand Audio Vumeter StopAll

Stop collecting VU meter information for all connectors.

Requires user role: `USER`

Example:

```
xCommand Audio Vumeter StopAll
*r AudioVumeterStopResult (status=OK):
** end
```

The Bookings commands

xCommand Bookings List

List the stored bookings for the system. The list of booking details is received from the management system. All parameters are optional, and can be used to limit the search result.

If no parameters are used, past, present and future bookings are all listed. To avoid listing bookings from yesterday and before, use DayOffset = 0.

Days: Number of days to retrieve bookings from.

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

Limit: Max number of bookings to list.

Offset: Offset number of bookings for this search.

Requires user role: `USER`

Parameters:

Days: <1..365>

DayOffset: <0..365>

Limit: <1..65534>

Offset: <0..65534>

Example:

```
Example 1 (One booking in list)
xCommand Bookings List
OK
*r Bookings ResultInfo TotalRows: 1
*r Bookings LastUpdated: "2011-09-02T11:19:01Z"
*r Bookings Booking 1 Id: "273"
*r Bookings Booking 1 Title: "Sales meeting"
*r Bookings Booking 1 Agenda: "Describe this command"
*r Bookings Booking 1 Privacy: Public
*r Bookings Booking 1 Organizer FirstName: "Ola"
*r Bookings Booking 1 Organizer LastName: "Normann"
*r Bookings Booking 1 Organizer Email: "ola.normann@domain.com"
*r Bookings Booking 1 Time StartTime: "2011-09-02T13:00:00Z"
*r Bookings Booking 1 Time StartTimeBuffer: 600
*r Bookings Booking 1 Time EndTime: "2011-09-02T13:30:00Z"
*r Bookings Booking 1 Time EndTimeBuffer: 0
*r Bookings Booking 1 MaximumMeetingExtension: 30
*r Bookings Booking 1 MeetingExtensionAvailability: Guaranteed
*r Bookings Booking 1 BookingStatus: OK
*r Bookings Booking 1 BookingStatusMessage: ""
*r Bookings Booking 1 Webex Enabled: True
*r Bookings Booking 1 Webex Url: "http://webex.url"
*r Bookings Booking 1 Webex MeetingNumber: "webexNumber@cisco.com"
*r Bookings Booking 1 Webex Password: ""
*r Bookings Booking 1 Webex HostKey: ""
*r Bookings Booking 1 Webex DialInNumbers DialInNumber 1 Type: TollFree
*r Bookings Booking 1 Webex DialInNumbers DialInNumber 1 Number: "+1 987
654321"
*r Bookings Booking 1 Webex DialInNumbers DialInNumber 2 Type: Toll
*r Bookings Booking 1 Webex DialInNumbers DialInNumber 2 Number: "+1 987
654322"
*r Bookings Booking 1 InteropBridge Number: ""
*r Bookings Booking 1 InteropBridge ConferenceId: ""
*r Bookings Booking 1 ManualCallIn Number: ""
*r Bookings Booking 1 ManualCallIn ConferenceId: ""
*r Bookings Booking 1 ManualCallIn ConferencePassword: ""
*r Bookings Booking 1 Encryption: BestEffort
*r Bookings Booking 1 Role: Slave
*r Bookings Booking 1 Recording: Disabled
*r Bookings Booking 1 DialInfo Calls Call 1 Number: "91123456;conference-
id=2100170569"
*r Bookings Booking 1 DialInfo Calls Call 1 Protocol: SIP
*r Bookings Booking 1 DialInfo Calls Call 1 CallRate: 3000
*r Bookings Booking 1 DialInfo Calls Call 1 CallType: Video
*r Bookings Booking 1 DialInfo ConnectMode: OBTP
** end

Example 2 (When no bookings have been received, or after all bookings have
been removed by the command xCommand Bookings Clear)
xCommand Bookings List
OK
*r Bookings Error: "No bookings found."
*r Bookings ResultInfo TotalRows: 0
*r Bookings LastUpdated: Never
** end
```

xCommand Bookings Clear

Clear the current stored list of bookings.

Requires user role: `USER`

Example:

```
xCommand Bookings Clear
** end
OK
```

The Boot commands

xCommand Boot

Reboot system.

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

Requires user role: `USER`

Parameters:

Action: <Restart/Shutdown>

Example:

```
xCommand Boot
*r BootResult (status=OK):
** end
OK
CUIL reboot request, restarting
Connection closed by foreign host.
```

The Call commands

xCommand Call Accept

Accept an incoming call. If no CallId is specified, all incoming calls will be accepted. The CallID is returned when the xCommand Dial command is run. During the call you can run the xStatus Call command to see the CallId.

Requires user role: `USER`

Parameters:

CallId: <0..65534>

Example:

```
xCommand Call Accept CallId:19
OK
*r CallAcceptResult (status=OK):
** end
```

xCommand Call Reject

Reject incoming call. If no call id is specified, all incoming calls will be rejected. The CallID is returned when the xCommand Dial command is run. During the call you can run the xStatus Call command to see the CallId.

Requires user role: `USER`

Parameters:

CallId: <0..65534>

Example:

```
xCommand Call Reject CallId:20
OK
*r CallRejectResult (status=OK):
** end
```

xCommand Call Disconnect

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

Requires user role: `USER`

Parameters:

CallId(r): <0..65534>

Example:

```
xCommand Call Disconnect CallId:17
OK
*r DisconnectCallResult (status=OK):
** end
```

xCommand Call DisconnectAll

Disconnect all active calls.

Requires user role: `USER`

Example:

```
xCommand Call DisconnectAll
OK
*r DisconnectAllResult (status=OK):
** end
```

xCommand Call Hold

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

Requires user role: `USER`

Parameters:

CallId(r): <0..65534>

Example:

```
xCommand Call Hold CallId:19
OK
*r CallHoldResult (status=OK):
** end
```

xCommand Call Join

Join all existing calls, active and on hold.

Requires user role: `USER`

Example:

```
xCommand Call Join
OK
*r CallJoinResult (status=OK):
** end
```

xCommand Call Resume

Resume a call that have been put on hold. The CallID is returned when the xCommand Dial command is run. During the call you can run the xStatus Call command to see the CallId.

Requires user role: `USER`

Parameters:

CallId(r): <0..65534>

Example:

```
xCommand Call Resume CallId:19
OK
*r CallResumeResult (status=OK):
** end
```

xCommand Call ExtendConference

Request to extend the conference for the call with the specified CallId.

For this command to apply, the following are required:

1) The result from the xStatus Conference Site <CallId> ConferenceExtended must be NotExtended:
*s Conference Site <CallId> ConferenceExtended: NotExtended

2) The following tag from the bookings list command must have a value > 0:

*r Bookings Booking <bookingNumber> MaximumMeetingExtension: "30"

Note: The command returns the result from the command itself, not the result from the extension request to far end. The result from the extension request will be given by a separate event.

CallId: CallId for the call in question.

Requires user role: `USER`

Parameters:

CallId(r): <0..65534>

Example:

```
xCommand Call ExtendConference CallId: 1
OK
*r CallExtendConferenceResult (status=OK)
** end
```

The CallLog commands

xCommand CallLog Clear

Clear call logs stored in the system. If a logtag is given as argument, that specific call is deleted from the logs. If no logtag is given then all call logs will be deleted. The LogTag values for the calls are found by issuing the xHistory CallLog Call command.

Requires user role: `USER`

Parameters:

LogTag: <0..2147483647>

Example:

```
xCommand CallLog Clear
*r ClearResult (status=OK):
** end
```

xCommand CallLog Recent Delete

Delete the call log of recent calls. If a logtag is given as argument, that specific call is deleted from the log. If no logtag is given, the complete recent calls log will be deleted. The LogTag values for recent calls are found by issuing the xHistory CallLog Recent command.

Requires user role: `USER`

Parameters:

LogTag: <0..2147483647>

Example:

```
xCommand CallLog Recent Delete
*r DeleteResult (status=OK):
** end
```

xCommand CallLog Outgoing Delete

Delete the call log of outgoing calls. If a logtag is given as argument, that specific call is deleted from the log. If no logtag is given, the complete outgoing calls log will be deleted. The LogTag values for outgoing calls are found by issuing the xHistory CallLog Outgoing command.

Requires user role: `USER`

Parameters:

LogTag: <0..2147483647>

Example:

```
xCommand CallLog Outgoing Delete LogTag:202
*r DeleteResult (status=OK):
** end
```

xCommand CallLog Received Delete

Delete the call log of received calls. If a logtag is given as argument, that specific call is deleted from the log. If no logtag is given, the complete received calls log will be deleted. The LogTag values for received calls are found by issuing the xHistory CallLog Received command.

Requires user role: `USER`

Parameters:

LogTag: <0..2147483647>

Example:

```
xCommand CallLog Received Delete LogTag:126
*r DeleteResult (status=OK):
** end
```

xCommand CallLog Missed Delete

Delete the call log of missed calls. If a logtag is given as argument, that specific call is deleted from the log. If no logtag is given, the complete missed calls log will be deleted. The LogTag values for missed calls are found by issuing the xHistory CallLog Missed command.

Requires user role: `USER`

Parameters:

LogTag: <0..2147483647>

Example:

```
xCommand CallLog Missed Delete LogTag:119
*r DeleteResult (status=OK):
** end
```

xCommand CallLog Missed Dismiss

Review the call log of dismissed calls. If a logtag is given as argument, that specific call is deleted from the log. If no logtag is given, the complete missed calls log will be deleted. The LogTag values for missed calls are found by issuing the xHistory CallLog Missed command.

Requires user role: `USER`

Parameters:

LogTag: <0..2147483647>

Example:

```
xCommand CallLog Missed Dimissed LogTag:119
*r DismissResult (status=OK):
** end
```

The CamCtrlPip commands

xCommand CamCtrlPip

Show or hide the camera selfview in a small window (picture in picture).

Requires user role: `USER`

Parameters:

Mode(r): <On/Off>

Example:

```
xCommand CamCtrlPip Mode: On
*r CamCtrlPipResult (status=OK):
** end
```

The Camera commands

xCommand Camera PanTiltReset

The camera is reset to its default values for pan and tilt. If the camera is daisy chained, the CameraId is given by its place in the chain.

Requires user role: `USER`

Parameters:

CameraId(r): <1..7>

Example:

```
xCommand Camera PanTiltReset CameraId:1
OK
*r PanTiltResetResult (status=OK):
** end
```

xCommand Camera PositionReset

Reset the camera position the the deafult position.

Requires user role: `USER`

Parameters:

CameraId(r): <1..7>

Example:

```
xCommand Camera PositionReset CameraId:1
OK
*r CameraPositionResetResult (status=OK):
** end
```

xCommand Camera PositionSet

Position the camera by defining the pan, tilt, zoom and focus parameters. If the camera is placed in a daisy chain you need to know the CameraId for the camera you want to address.

Requires user role: `USER`

Parameters:

CameraId(r): <1..7>

Pan: <-65535..65535>

Tilt: <-65535..65535>

Zoom: <0..65535>

Focus: <0..65535>

Example:

```
xCommand Camera PositionSet CameraId:1 Pan:200 Tilt:300
OK
*r CameraPositionSetResult (status=OK):
** end
```

xCommand Camera Ramp

Move the camera in a specified direction. The camera will move at specified speed until a stop command is issued. In a daisy chain, you need to know the CameraId for the camera you want to address. Be aware that pan and tilt can be operated simultaneously, but no other combinations. In the latter case only the first operation specified will be executed. For example, if you try to run both zoom and pan at the same time, only zoom is executed.

NOTE: You must run a stop command to stop the camera, see the example below.

CameraId: Give the camera id.

Pan: Move the camera to the Left or Right, followed by Stop.

PanSpeed: Set the pan speeed.

Tilt: Move the camera Up or Down, followed by Stop.

TiltSpeed: Set the tilt speeed.

Zoom: Zoom the camera In or Out, followed by Stop.

ZoomSpeed: Set the zoom speeed.

Focus: Focus the camera Far or Near, followed by Stop.

Requires user role: `USER`

Parameters:

CameraId(r): <1..7>

Pan: <Left/Right/Stop>

PanSpeed: <1..15>

Tilt: <Down/Up/Stop>

TiltSpeed: <1..15>

Zoom: <In/Out/Stop>

ZoomSpeed: <1..15>

Focus: <Far/Near/Stop>

Example:

```
xCommand Camera Ramp CameraId:1 Pan:left PanSpeed:1
OK
*r RampResult (status=OK):
** end
xCommand Camera Ramp CameraId:1 Pan:stop
OK
*r RampResult (status=OK):
** end
```

xCommand Camera ReconfigureCameraChain

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

Requires user role: `USER`

Example:

```
xCommand Camera ReconfigureCameraChain
*r ReconfigureCameraChainResult (status=OK):
** end
```

xCommand Camera TriggerAutofocus

Trigger the autofocus functionality. The camera must support autofocus functionality. If the camera is daisy chained, the CameraId is given by its place in the chain.

Requires user role: `USER`

Parameters:

CameraId(r): <1..7>

Example:

```
xCommand Camera TriggerAutofocus CameraId:1
OK
*r TriggerAutofocusResult (status=OK):
** end
```

xCommand Camera PositionActivateFromPreset

Selects pan, tilt, zoom and focus parameters for the given camera id from the selected preset.

CameraId: Give the camera id.

PresetId: Select preset 1 to 15.

Requires user role: `USER`

Parameters:

CameraId(r): <1..7>

PresetId(r): <1..15>

Example:

```
xCommand Camera PositionActivateFromPreset CameraId:1 PresetID:1
OK
*r PositionActivateFromPresetResult (status=OK):
** end
```

xCommand Camera Preset Activate

This command activates a stored camera preset.

Note: The xCommand Camera Preset commands are used to store camera positions for individual cameras. This is in contrast to the xCommand Preset commands where a single preset stores/recalls ALL connected cameras plus the Video Input switcher settings. This makes it more usable for integrations where one wants to store multiple camera positions individually per camera rather than a complete camera position set.

PresetId: The preset number you want to activate.

Requires user role: `USER`

Parameters:

PresetId(r): <1..35>

Example:

```
xCommand Camera Preset Activate PresetId: 1
OK
*r CameraPresetActivateResult (status=OK)
** end
```

xCommand Camera Preset Edit

Edit a stored camera preset.

Note: The xCommand Camera Preset commands are used to store camera positions for individual cameras. This is in contrast to the xCommand Preset commands where a single preset stores/recalls ALL connected cameras plus the Video Input switcher settings. This makes it more usable for integrations where one wants to store multiple camera positions individually per camera rather than a complete camera position set.

PresetId: The preset number you want to edit.

ListPosition: The sort order position in the xCommand Camera Preset List result.

Name: The name of the preset in the xCommand Camera Preset List result.

Requires user role: `USER`

Parameters:

PresetId(r): <1..35>

ListPosition: <1..35>

Name: <S: 0, 255>

Example:

```
xCommand Camera Preset Edit PresetId: 1 ListPosition: 1 Name: ""
OK
*r CameraPresetEditResult (status=OK)
** end
```

xCommand Camera Preset List

List information about available camera presets.

Note: The xCommand Camera Preset commands are used to store camera positions for individual cameras. This is in contrast to the xCommand Preset commands where a single preset stores/recalls ALL connected cameras plus the Video Input switcher settings. This makes it more usable for integrations where one wants to store multiple camera positions individually per camera rather than a complete camera position set.

PresetId: Filter on specified preset.

Requires user role: `USER`

Parameters:

PresetId: <1..35>

Example:

```
xCommand Camera Preset List PresetId: 1
OK
*r CameraPresetListResult (status=OK)
** end
```

xCommand Camera Preset Remove

Remove a camera preset.

Note: The xCommand Camera Preset commands are used to store camera positions for individual cameras. This is in contrast to the xCommand Preset commands where a single preset stores/recalls ALL connected cameras plus the Video Input switcher settings. This makes it more usable for integrations where one wants to store multiple camera positions individually per camera rather than a complete camera position set.

PresetId: The id of the camera preset to remove.

Requires user role: `USER`

Parameters:

PresetId(r): <1..35>

Example:

```
xCommand Camera Preset Remove PresetId: 1
OK
*r CameraPresetRemoveResult (status=OK)
** end
```

xCommand Camera Preset Store

Add a new camera preset.

Note: The xCommand Camera Preset commands are used to store camera positions for individual cameras. This is in contrast to the xCommand Preset commands where a single preset stores/recalls ALL connected cameras plus the Video Input switcher settings. This makes it more usable for integrations where one wants to store multiple camera positions individually per camera rather than a complete camera position set.

PresetId: Optional Id for this preset.

CameraId: Which camera to store the position of.

ListPosition: The sort order of the new preset.

Name: The name that will be used in the listing of presets.

Requires user role: `USER`

Parameters:

PresetId: <1..35>

CameraId: <1..7>

ListPosition: <1..35>

Name: <S: 0, 255>

Example:

```
xCommand Camera Preset Store PresetId: 1 CameraId: 1 ListPosition: 1 Name ""
OK
*r CameraPresetStoreResult (status=OK)
** end
```

The Conference commands

xCommand Conference DoNotDisturb Activate

This command switches on Do Not Disturb, and the Timeout parameter allows you to control when it is switched off again. It will only take effect when xConfiguration Conference DoNotDisturb Mode is set to Timed.

When Do Not Disturb is switched on, all incoming calls will be rejected and they will be registered as missed calls. The calling side will receive a busy signal.

Timeout: The number of minutes before Do Not Disturb is switched off. If not set, Do Not Disturb times out after 1440 minutes.

Requires user role: `USER`

Parameters:

Timeout: <0..1440>

Example:

```
xCommand Conference DoNotDisturb Activate
*r ActivateResult:
** end
```

xCommand Conference DoNotDisturb Deactivate

TThis command switches off Do Not Disturb. This command will only take effect when xConfiguration Conference DoNotDisturb Mode is set to Timed.

When Do Not Disturb is switched off incoming calls will come through as normal.

Requires user role: `USER`

Example:

```
xCommand Conference DoNotDisturb Deactivate
*r DeactivateResult:
** end
```

The Dial commands

xCommand Dial

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

Number: Enter the number or address.

Protocol: Select the H323 or SIP protocol.

CallRate: Set a call rate.

CallType: Select the audio or video call type.

BookingId: Any identifier that an external booking system (e.g. TMS, CTS-MAN) can use for its own references to match placed calls with the booking systems internal identifier for a meeting. This can be any string, e.g. a GUID. The booking Id will be supplied in call logs, call events etc for the call.

Requires user role: `USER`

Parameters:

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

Protocol: <H323/Sip>

CallRate: <64..6000>

CallType: <Audio/Video>

BookingId: <S: 0, 255>

Example:

```
xCommand Dial Number:543210 Protocol:h323
OK
*r DialResult (status=OK):
CallId: 2
ConferenceId: 1
** end
```

The DTMFSend commands

xCommand DTMFSend

Send DTMF tones to the far end.

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

DTMFString: Enter the DTMF string.

Requires user role: `USER`

Parameters:

CallId: <0..65534>

DTMFString(r): <S: 0, 32>

Example:

```
xCommand DTMFSend CallId:2 DTMFString:1234
*r DTMFSendResult (status=OK):
** end
```

The FacilityService commands

xCommand FacilityService Dial

Dial out from the system to a facility service. A maximum of five facility services can be defined; which one of these five to dial to is identified by the ServiceId (ref. xConfiguration Facility Service [ServiceId] Type/Name/Number/CallType). The command returns information about the CallId and ConferenceId.

ServiceId: The identifier of the facility service.

Requires user role: `USER`

Parameters:

ServiceId(r): <1..5>

Example:

```
xCommand FacilityService Dial ServiceId: 1
OK
*r FacilityServiceDialResult (status=OK):
   CallId: 2
   ConferenceId: 1
** end
```

The FarEndControl commands

xCommand FarEndControl Camera Move

Move the far end camera (the remote camera). NOTE: The far end camera will move in the specified direction until the stop command (ref: xCommand FarEndControl Camera Stop) is issued

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

Value: Select the action for how to move the camera.

Requires user role: `USER`

Parameters:

CallId: <0..65534>

Value(r): <Left/Right/Up/Down/ZoomIn/ZoomOut>

Example:

```
xCommand FarEndControl Camera Move CallId:3 Value:left
*r FECCMoveResult (status=OK):
** end
```

xCommand FarEndControl Camera Stop

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

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

Requires user role: `USER`

Parameters:

CallId: <0..65534>

Example:

```
xCommand FarEndControl Camera Stop CallId:3
*r FECCMoveResult (status=OK):
** end
```

xCommand FarEndControl Preset Activate

Move the far end camera to a camera preset position.

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

PresetId: The PresetId for far end control must be retrieved from the far end codec.

Requires user role: `USER`

Parameters:

CallId: <0..65534>

PresetId(r): <1..15>

Example:

```
xCommand FarEndControl Preset Activate CallId:3 PresetId:1
*r FECCPresetActivateResult (status=OK):
** end
```

xCommand FarEndControl Preset Store

Store the far end camera position to a camera preset.

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

PresetId: The PresetId for far end control must be retrieved from the far end codec.

Requires user role: `USER`

Parameters:

CallId: <0..65534>

PresetId(r): <0..15>

Example:

```
xCommand FarEndControl Preset Store CallId:3 PresetId:1
*r FECCPresetStoreResult (status=OK):
** end
```

xCommand FarEndControl Source Select

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

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

SourceId: Select a video input source on the far end.

Requires user role: `USER`

Parameters:

CallId: <0..65534>

SourceId(r): <0..15>

Example:

```
xCommand FarEndControl Source Select CallId:3 SourceId:1
*r FECCSelectSourceResult (status=OK):
** end
```

The GPIO commands

xCommand GPIO ManualState Set

NOTE: This command is not supported on Codec C40.

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

Requires user role: `USER`

Parameters:

Pin1: <High/Low>

Pin2: <High/Low>

Pin3: <High/Low>

Pin4: <High/Low>

Example:

```
xCommand GPIO ManualState Set Pin1: High
*r GpioStateSetResult (status=OK):
** end
```

The HttpFeedback commands

xCommand HttpFeedback Register

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

FeedbackSlot: You can have from 1 to 4 slots for feedback.

ServerUrl: Define the URL for the HTTP(S) server.

Expression[1..15]: What parts of the Status and Configuration XML documents to monitor are specified by XPath expressions. You can have from 1 to 15 XPath expressions.

Requires user role: `ADMIN`

Parameters:

FeedbackSlot: <1..4>

ServerUrl(r): <S: 1, 2048>

Expression[1..15]: <S: 1, 255>

Example:

```
xCommand HttpFeedback Register FeedbackSlot:1 ServerUrl:10.47.19.41
Expression[1]:Status/Video Expression[2]:Status/Audio Expression[3]:Status/Call
Expression[4]:Status/Conference
OK
*r FeedbackRegisterResult (status=OK):
FeedbackSlot: 1
** end
```

xCommand HttpFeedback Deregister

Deregister XML feedback over HTTP(S).

Requires user role: `ADMIN`

Parameters:

FeedbackSlot(r): <1..4>

Example:

```
xCommand HttpFeedback Deregister FeedbackSlot:1
OK
*r FeedbackDeregisterResult (status=OK):
FeedbackSlot: 1
** end
```

The Key commands

xCommand Key Click

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

Requires user role: `ADMIN`

Parameters:

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

Example:

```
xCommand Key Click Key:Down
*r KeyClickResult (status=OK):
** end
```

xCommand Key Press

Emulates a remote control key press without releasing it. The Key Press command must be followed by a Key Release command to emulate releasing the key.

Requires user role: `ADMIN`

Parameters:

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

Example:

```
xCommand Key Press Key:Home
*r KeyPressResult (status=OK):
** end
```

xCommand Key Release

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

Requires user role: `ADMIN`

Parameters:

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

Example:

```
xCommand Key Release Key:Home
*r KeyReleaseResult (status=OK):
** end
```

The Message commands

xCommand Message Alert Display

Display a message on screen, for a specified duration of time (in seconds). NOTE: If Duration is not set, the command must be followed by xCommand Message Alert Clear.

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

Title: Enter a message title.

Text: Enter the message to be displayed.

Duration: Set how long (in seconds) the message is to be displayed on the screen. If set to 0 (zero) the message will not disappear until a xCommand Message Alert Clear message has been sent.

Requires user role: `ADMIN`

Parameters:

Title: <S: 0, 255>

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

Duration: <0..3600>

Example:

```
xCommand Message Alert Display Title: "Message" Text: "The meeting will end
in 5 minutes." Duration: 20
OK
*r MessageAlertDisplayResult (status=OK):
** end
```

xCommand Message Alert Clear

Remove the message which was displayed using the xCommand Message Alert Display command. This is required when the Duration parameter is not set.

Requires user role: `ADMIN`

Example:

```
xCommand Message Alert Clear
OK
*r MessageAlertClearResult (status=OK):
** end
```

xCommand Message TextLine Display

Display a text line on screen. Optionally you can place the text line at a specified location and for a specified duration of time (in seconds). NOTE: If Duration is not set, the command must be followed by xCommand Message TextLine Clear.

Text: Enter the text line to be displayed.

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

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

Duration: Set how long (in seconds) the text line is to be displayed on the screen. If set to 0 (zero) the text line will be displayed until a xCommand Message TextLine Clear command has been sent.

Requires user role: `ADMIN`

Parameters:

Text(r): <S: 0, 140>

X: <1..10000>

Y: <1..10000>

Duration: <0..3600>

Example:

```
xCommand Message TextLine Display Text: "The message to be displayed." X:500
Y:2000 Duration: 20
OK
*r MessageTextLineDisplayResult (status=OK):
** end
```

xCommand Message TextLine Clear

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

Requires user role: `ADMIN`

Example:

```
xCommand Message TextLine Clear
OK
*r MessageTextLineClearResult (status=OK):
** end
```

xCommand Message Prompt Display

Display a small window on screen with a title, text and up to five options for response from the user. The message will display on screen until the user gives a response, or until the system receives the following command xCommand Message Prompt Clear.

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

Title: Enter the title of the message.

Text: Enter the message.

FeedbackId: To identify the feedback enter a FeedbackId.

Option.1 to Option.5: Enter the text to appear on the feedback options.

Requires user role: `ADMIN`

Parameters:

Title: <S: 0, 255>

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

FeedbackId: <S: 0, 255>

Option.1: <S: 0, 255>

Option.2: <S: 0, 255>

Option.3: <S: 0, 255>

Option.4: <S: 0, 255>

Option.5: <S: 0, 255>

Example:

```
xCommand Message Prompt Display Title: "Meeting extension" Text: "The
meeting is about to end. Do you want to extend the meeting?" Option.1:
"No" Option.2: "Yes, extend with 5 minutes" Option.3: "Yes, extend with 10
minutes"
OK
*r MessagePromptDisplayResult (status=OK):
** end
```

xCommand Message Prompt Clear

Remove the window displayed using the xCommand Message Alert Display command.

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

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

Requires user role: `ADMIN`

Parameters:

FeedbackId: <S: 0, 255>

Example:

```
xCommand Message Prompt Clear
OK
*r MessagePromptClearResult (status=OK):
** end
```

xCommand Message Prompt Response

Give a response to the xCommand Message Prompt Display.

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

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

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

Requires user role: `ADMIN`

Parameters:

FeedbackId: <S: 0, 255>

OptionId(r): <1..5>

Example:

```
xCommand Message Prompt Response OptionId: 1
OK
*r MessagePromptResponseResult (status=OK):
** end
```

xCommand Message Echo

Issuing the command will make the API raise a message-echo event. The command has no other impact on the codec. Usage can be to poll the codec from a control system or any external device/system to check for connectivity. To monitor the feedback use the xFeedback command. You can read more about the xFeedback command in the general API introduction section.

Text: Enter the text to be echoed.

Requires user role: `ADMIN`

Parameters:

Text: <S: 0, 255>

Example:

```
xCommand Message Echo Text:" MyEchoListner99"
*** end
* e Message Echo Text: "MyEchoListner99"
```

The Phonebook commands

xCommand Phonebook Folder Add

Add a folder to the local phonebook, where phonebook entries can be stored. Returns the FolderId (localGroupId-3), which is a unique Id of the folder.

Name(r): The name of the folder.

ParentFolderId: A unique identifier for the parent folder, which was created when a previous xCommand Phonebook Folder Add command was issued.

Requires user role: `ADMIN`

Parameters:

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

ParentFolderId: <S: 0, 255>

Example:

```
xCommand Phonebook Folder Add Name: "New York Office"
OK
*r PhonebookFolderAddResult (status=OK):
 Name: localGroupId-3
** end
```

xCommand Phonebook Folder Modify

Modify an existing phonebook folder.

FolderId: A unique identifier for the folder, which was created when the xCommand Phonebook Folder Add command was issued.

Name(r): The name of the contact.

ParentFolderId: A unique identifier for the parent folder, which was created when the xCommand Phonebook Folder Add command was issued.

Requires user role: `ADMIN`

Parameters:

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

Name: <S: 0, 255>

ParentFolderId: <S: 0, 255>

Example:

```
xCommand Phonebook Folder Modify FolderId:localGroupId-3 Name: "New York
Head Office"
OK
*r PhonebookFolderModifyResult (status=OK):
** end
```

xCommand Phonebook Folder Delete

Delete an existing folder from the local phonebook.

FolderId: A unique identifier for the folder, which was created when the xCommand Phonebook Folder Add command was issued.

Requires user role: `ADMIN`

Parameters:

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

Example:

```
xCommand Phonebook Folder Delete FolderId:localGroupId-3
OK
*r PhonebookFolderDeleteResult (status=OK):
** end
```

xCommand Phonebook Contact Add

Add new contact to the local phonebook. Stored internally in the system. Returns the ContactId (Name: localContactId-1), which is a unique Id of the contact.

Name: The name of the contact.

FolderId: A unique identifier for the folder, which was created when the xCommand Phonebook Folder Add command was issued.

ImageURL: The URL to an image.

Title: The title of the contact.

Number: The phone number or address of the contact.

Protocol: Select H323 or SIP protocol.

CallRate: Set a call rate.

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

Device: Select the device type.

Requires user role: `ADMIN`

Parameters:

Name(r): <S: 0, 255>
FolderId: <S: 0, 255>
ImageURL: <S: 0, 255>
Title: <S: 0, 255>
Number: <S: 0, 255>
Protocol: <H323/SIP>
CallRate: <0..65534>
CallType: <Audio/Video>
Device: <Mobile/Other/Telephone/Video>

Example:

```
xCommand Phonebook Contact Add Name: "John Doe" Number:12345
OK
*r PhonebookContactAddResult (status=OK):
Name: localContactId-1
** end
```

xCommand Phonebook Contact Modify

Modify the contact details of an existing contact in the local phonebook.

ContactId: A unique identifier for the contact, which was created when the xCommand Phonebook Contact Add command was issued.

Name: The name of the contact.

FolderId: A unique identifier for the folder, which was created when the xCommand Phonebook Folder Add command was issued.

ImageURL: The URL to an image.

Title: The title of the contact.

Requires user role: `ADMIN`

Parameters:

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

Name: <S: 0, 255>

FolderId: <S: 0, 255>

ImageURL: <S: 0, 255>

Title: <S: 0, 255>

Example:

```
xCommand Phonebook Contact Modify ContactId:localContactId-1 Name: "John Doe
- office"
OK
*r PhonebookContactModifyResult (status=OK):
** end
```

xCommand Phonebook Contact Delete

Delete an existing contact from local phonebook.

ContactId: A unique identifier for the contact, which was created when the xCommand Phonebook Contact Add command was issued.

Requires user role: `ADMIN`

Parameters:

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

Example:

```
xCommand Phonebook Contact Delete ContactId:localContactId-1
OK
*r PhonebookContactDeleteResult (status=OK):
** end
```

xCommand Phonebook ContactMethod Add

Add details about the call setup to an existing contact in the local phonebook. Returns the ContactMethodId (Name: 1), which is a unique Id of the contact method.

ContactId: A unique identifier for the contact, which was created when the xCommand Phonebook Contact Add command was issued.

Device: Set which type of device to call to.

Number(r): The phone number or address of the contact.

Protocol: Select H323 or SIP protocol.

CallRate: Set a call rate.

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

Requires user role: `ADMIN`

Parameters:

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

Device: <Mobile/Other/Telephone/Video>

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

Protocol: <H323/SIP>

CallRate: <0..65534>

CallType: <Audio/Video>

Example:

```
xCommand Phonebook ContactMethod Add ContactId:localContactId-2 Number:54321
Protocol:H323
OK
*r PhonebookContactMethodAddResult (status=OK):
Name: 1
** end
```

xCommand Phonebook ContactMethod Modify

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

ContactId: A unique identifier for the contact. It was created when the xCommand Phonebook Contact Add command was issued.

ContactMethodId: A unique identifier for the contact method. It was created when the xCommand Phonebook ContactMethod Add command was issued.

Device: Set which type of device to call to.

Number: The phone number or address of the contact.

Protocol: Select H323 or SIP protocol.

CallRate: Set a call rate.

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

Requires user role: `ADMIN`

Parameters:

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

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

Device: <Mobile/Other/Telephone/Video>

Number: <S: 0, 255>

Protocol: <H323/SIP>

CallRate: <0..65534>

CallType: <Audio/Video>

Example:

```
xCommand Phonebook ContactMethod Modify ContactMethodId:117
ContactId:localContactId-10 Number:"newnumber@cisco.com"
OK
```

xCommand Phonebook ContactMethod Delete

Delete details about the call setup to an existing contact in the local phonebook.

ContactId: A unique identifier for the contact, which was created when the xCommand Phonebook Contact Add command was issued.

ContactMethodId: A unique identifier for the contact method, which was created when the xCommand Phonebook ContactMethod Add command was issued.

Requires user role: `ADMIN`

Parameters:

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

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

Example:

```
xCommand Phonebook ContactMethod Delete ContactId:localContactId-2
ContactMethodId:1
OK
*r PhonebookContactMethodDeleteResult (status=OK):
** end
```

xCommand Phonebook Search

The search command lets you search in both the local and corporate phone books. A search will give a ResultSet. More examples can be found on the Developer Zone web page. Go to: http://developer.tandberg.com/web/guest/howtos/cseries-api/phonebook.

PhonebookId: The value of the ID tag for which phonebook server to use. See xConfiguration Phonebook Server. Not necessary to use.

PhonebookType: Which phone book to search in. Either the local phone book or the corporate phonebook.

SearchString: Search for entries containing specified string (not begins with). If no FolderId is specified, the search will yield search results from ALL folders/phonebook directories. The SearchString parameter is optional for software version TC2.0 and later.

SearchField: Currently not in use.

Offset: Get records starting with this offset in a search. Default 0. Used together with Limit to support paging.

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

Limit: Limit the number of records in the result set to this number. E.g. Limit: 10 will only give a ResultSet of 10 entries (Contacts + Folders) although the total number of hits may be greater.

Recursive: Set if the phonebook should search recursive. The result from an empty search will return both the directories and the content in the directories. NOTE: This command is only valid for the local directory.

Requires user role: `USER`

Parameters:

PhonebookId: <S: 0, 255>

PhonebookType: <Corporate/Local>

SearchString: <S: 0, 255>

SearchField: <Name/Number>

Offset: <0..65534>

FolderId: <S: 0, 255>

Limit: <0..65534>

Recursive: <False/True>

Example:

```
xCommand Phonebook Search PhonebookType:Corporate Limit:2
FolderId:"corporate_001"
OK
*r ResultSet ResultInfo TotalRows: 25
*r ResultSet Contact 1 Name: "/tmp"
*r ResultSet Contact 1 ContactId: "e_9664921"
*r ResultSet Contact 1 ContactMethod 1 ContactMethodId: "1"
*r ResultSet Contact 1 ContactMethod 1 Number: "H323:/John.Doe.Office@company.
```

```
com"
*r ResultSet Contact 1 ContactMethod 1 Protocol: H323
*r ResultSet Contact 1 ContactMethod 2 ContactMethodId: "2"
*r ResultSet Contact 1 ContactMethod 2 Number: "SIP:/John.Doe.Office@company.
com"
*r ResultSet Contact 1 ContactMethod 2 Protocol: SIP
*r ResultSet Contact 2 Name: "Other"
*r ResultSet Contact 2 ContactId: "e_9468400"
*r ResultSet Contact 2 ContactMethod 1 ContactMethodId: "3"
*r ResultSet Contact 2 ContactMethod 1 Number: "H323:John.Doe.Home@company.
com"
*r ResultSet Contact 2 ContactMethod 1 Protocol: H323
*r ResultSet Contact 2 ContactMethod 2 ContactMethodId: "4"
*r ResultSet Contact 2 ContactMethod 2 Number: "SIP:John.Doe.Home@company.
com"
*r ResultSet Contact 2 ContactMethod 2 Protocol: SIP
** end
```

The Presentation commands

xCommand Presentation Start

Open a media stream from the selected presentation source.

PresentationSource: Select the video input source to be used for presentation.

Requires user role: `USER`

Parameters:

PresentationSource: <1..3>

Example:

```
xCommand Presentation Start PresentationSource:2
OK
*r PresentationStartResult (status=OK):
** end
```

xCommand Presentation Stop

Stop the media stream from the presentation source.

Requires user role: `USER`

Example:

```
xCommand Presentation Stop
OK
*r PresentationStopResult (status=OK):
** end
```

The Preset commands

xCommand Preset Store

Store the connector selections for all video input sources and the current camera position for all cameras. Note that all video input sources and all camera positions are included in each preset. The system may hold 15 predefined video input presets.

PresetId: Select preset 1 to 15.

Type: Select Camera or All. Currently there is no difference if you select Camera or All.

Description: Enter a description of the camera preset.

Requires user role: `USER`

Parameters:

PresetId(r): <1..15>

Type(r): <All/Camera>

Description: <S: 0, 255>

Example:

```
xCommand Preset Store PresetId:3 Type:Camera Description:"Left view"
OK
*r PresetStoreResult (status=OK):
** end
```

xCommand Preset Activate

Activate one of the local presets.

Requires user role: `USER`

Parameters:

PresetId(r): <1..15>

Example:

```
xCommand Preset Activate PresetId:3
OK
*r PresetActivateResult (status=OK):
** end
```

xCommand Preset Clear

Delete an existing preset.

Requires user role: `USER`

Parameters:

PresetId(r): <1..15>

Example:

```
xCommand Preset Clear PresetId:3
OK
*r PresetClearResult (status=OK):
** end
```

The Provisioning commands

xCommand Provisioning StartUpgrade

The codec software can be upgraded from the provisioning server. When starting the upgrade the software is automatically downloaded and installed. The codec reboots to complete the software upgrade.

Requires user role: `USER`

Example:

```
xCommand Provisioning StartUpgrade
*r StartUpgradeResult (status=OK):
** end
```

The Security commands

xCommand Security FIPSMode Activate

Activate FIPS (140-2) mode. NOTE: Activating FIPS mode implies a reset to factory defaults.

While in FIPS mode, software upgrade is disabled and the following limitations will apply: All calls will be encrypted, unencrypted communication protocols like Telnet and HTTP cannot be used. IEEE802.1x and SNMP are disabled. The SIP Profile Type setting cannot be Microsoft. The root user is not available (root settings cannot be changed).

To exit FIPS mode, perform a factory reset.

Confirm: <Yes>

Requires user role: `ADMIN`

Parameters:

Confirm(r): <Yes>

Example:

```
xCommand Security FIPSMode Activate Confirm: "Yes"
OK
*r SecurityFIPSModeActivateConfirmResult (status=OK)
** end
```

The SStringSend commands

xCommand SStringSend

Send data to the far end, e.g. for control systems. Uses the H.224 data channel (UDP).

Message: Enter the message to be sent to the far end.

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

Requires user role: `ADMIN`

Parameters:

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

CallId: <0..65534>

Example:

```
xCommand SStringSend CallId:4 Message:"This is a test"
*r SStringSendResult (status=OK):
** end
```

The Standby commands

xCommand Standby Activate

Set the system in standby mode, which will turn off the video outputs and put the camera into sleep mode.

Requires user role: `USER`

Example:

```
xCommand Standby Activate
*r ActivateResult (status=OK):
** end
```

xCommand Standby Deactivate

Bring the system out of standby mode.

Requires user role: `USER`

Example:

```
xCommand Standby Deactivate
*r DeactivateResult (status=OK):
** end
```

xCommand Standby ResetTimer

Set a nonpermanent standby delay (in minutes). If the system is in standby mode when the reset timer is set, the system is brought out of standby mode. When left idle for the given delay the system goes into standby mode. Setting the reset timer will not affect the Standby Delay in the Advanced configuration menu (or by xConfiguration Standby Delay). Next time this delay will be the valid standby delay.

Requires user role: `USER`

Parameters:

Delay(r): <1..480>

Example:

```
xCommand Standby ResetTimer Delay:10
*r ResetResult (status=OK):
** end
```

The SystemUnit commands

xCommand SystemUnit OptionKey Add

Add an option key to support additional features.

Requires user role: `ADMIN`

Parameters:

Key(r): <S: 16, 24>

Example:

```
xCommand SystemUnit OptionKey Add Key:*****************
*r OptionKeyResult (status=OK):
** end
```

xCommand SystemUnit ReleaseKey Add

Add the software release key. Used to enable new software (applicable for main software releases).

Requires user role: `ADMIN`

Parameters:

Key(r): <S: 16, 24>

Example:

```
xCommand SystemUnit ReleaseKey Add Key:*****************
*r ReleaseKeyResult (status=OK):
** end
```

xCommand SystemUnit AdminPassword Set

Set an administrator password to restrict access the codec. After a restart of the codec this password will also apply to the web interface.

Requires user role: `USER`

Parameters:

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

Example:

```
xCommand SystemUnit AdminPassword Set Password:***********
*r AdminPasswordSetResult (status=OK):
** end
```

xCommand SystemUnit MenuPassword Set

Set a menu password to restrict access to Administrator Settings menu. If you have a remote control the password can also be set from the on screen menu. If you have a Cisco TelePresence Touch controller the menu password is set from the command line interface.

Requires user role: `USER`

Parameters:

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

Example:

```
xCommand SystemUnit MenuPassword Set Password:***********
*r MenuPasswordSetResult (status=OK):
** end
```

xCommand SystemUnit MenuPassword Validate

Validate that the supplied password is correct.

Requires user role: `USER`

Parameters:

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

Example:

```
xCommand SystemUnit MenuPassword Validate Password:***********
*r MenuPasswordValidateResult (status=OK):
** end
```

xCommand SystemUnit DateTime Set

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

Requires user role: `ADMIN`

Parameters:

Year: <2008..2037>

Month: <1..12>

Day: <1..31>

Hour: <0..23>

Minute: <0..59>

Second: <0..59>

Example:

```
xCommand SystemUnit DateTime Set Year:2009 Month:7 Day:3 Hour:12 Minute:0
Second:0
*r DateTimeSetResult (status=OK):
** end
```

xCommand SystemUnit DateTime Get

Read the time and date from the system.

Requires user role: `USER`

Example:

```
xCommand SystemUnit DateTime get
*r DateTimeGetResult (status=OK):
 Year: 2009
 Month: 7
 Day: 3
 Hour: 12
 Minute: 0
 Second: 0
** end
```

xCommand SystemUnit FactoryReset

Reset the codec to factory default settings. The call logs will be deleted and all system parameters will be reset to default values. All files that have been uploaded to the codec will be deleted. The Release key and Option key will not be affected.

As a default the system restarts after the factory reset, but other behavior can be forced by selecting a different TrailingAction.

TrailingAction: Select Shutdown or NoAction to override the default behavior (Restart).

Requires user role: `ADMIN`

Parameters:

Confirm(r): <Yes>

TrailingAction: <NoAction/Restart/Shutdown>

Example:

```
xCommand SystemUnit FactoryReset Confirm: Yes
*r FactoryResetConfirmResult (status=OK):
** end
```

xCommand SystemUnit SoftwareUpgrade

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

Requires user role: `USER`

Parameters:

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

UserName: <S: 0, 255>

Password: <S: 0, 255>

Example:

```
xCommand xCommand SystemUnit SoftwareUpgrade URL: "ftp://<ftp _ server _ ip _
address>/s52000tc4 _ 0 _ 0.pkg" UserName: testDownload Password: 1234
*r SystemUnitSoftwareUpgradeResult (status=OK):
** end
```

xCommand SystemUnit ConfigurationProfile Change

Select a previously saved configuration profile. Will be active after next system boot.

Requires user role: `USER`

Parameters:

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

Example:

```
xCommand SystemUnit ConfigurationProfile Change Name: "My _
ConfigurationProfile _ 1"
*r ConfigurationProfileChangeResult (status=OK):
   Warning: New configuration profile will be active after next boot.
** end
```

xCommand SystemUnit ConfigurationProfile Remove

Delete a configuration profile that has been stored in the system.

Requires user role: `USER`

Parameters:

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

Example:

```
xCommand SystemUnit ConfigurationProfile Remove Name: "My _
ConfigurationProfile _ 1"
*r ConfigurationProfileRemoveResult (status=OK):
** end
```

xCommand SystemUnit ConfigurationProfile SaveCurrentConfigurationAs

Save the current system settings into a configuration profile. Assign a name to the new profile. The name is the unique identifier of the profile.

Requires user role: `USER`

Parameters:

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

Example:

```
xCommand SystemUnit ConfigurationProfile SaveCurrentConfigurationAs Name:
"My _ ConfigurationProfile _ 1"
*r ConfigurationProfileSaveCurrentConfigurationResult (status=OK):
** end
```

xCommand SystemUnit ConfigurationProfile List

List configuration profiles that has been stored in the system.

Requires user role: `USER`

Example:

```
xCommand SystemUnit ConfigurationProfile List
*r ConfigurationProfileListResult (status=OK):
   Profile: My _ ConfigurationProfile _ 1
   Profile: My _ ConfigurationProfile _ 2
** end
```

xCommand SystemUnit ConfigurationProfile CancelChange

Cancel the "ConfigurationProfile Change" command, that would otherwise take effect after next system boot.

Requires user role: `USER`

Example:

```
xCommand SystemUnit ConfigurationProfile CancelChange
*r ConfigurationProfileCancelChangeResult (status=OK):
** end
```

xCommand SystemUnit Notifications RemoveAll

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

Requires user role: `USER`

Example:

```
xCommand SystemUnit Notifications RemoveAll
*r NotificationsRemoveAllResult (status=OK):
** end
```

xCommand SystemUnit ReleaseKey List

List all software versions that the system has a valid release key for.

Requires user role: `USER`

Example:

```
xCommand SystemUnit ReleaseKey List
*r SystemUnit ReleaseKey 1 Version: "TC1"
*r SystemUnit ReleaseKey 2 Version: "TC2"
*r SystemUnit ReleaseKey 3 Version: "TC3"
*r SystemUnit ReleaseKey 4 Version: "TC4"
*r SystemUnit ReleaseKey 5 Version: "TC5"
** end
```

The TStringSend commands

xCommand TStringSend

Send data to far end, e.g. for Telepresence control systems. Uses the H.245 control channel (TCP). Works with H.323 calls only.

Message: Enter the message to be sent to the far end.

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

Requires user role: `USER`

Parameters:

CallId(r): <0..65534>

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

Example:

```
xCommand TStringSend CallId:1 Message:"This is an example"
*r TStringSendResult (status=OK):
** end
```

The Video commands

xCommand Video PictureLayoutSet

Select the screen layout mode.

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

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

LayoutFamily: Select a layout family.

CustomLayoutName: Enter a name for the layout.

Requires user role: `USER`

Parameters:

Target: <local/remote>

CallId: <0..65534>

LayoutFamily(r): <auto/custom/equal/fullscreen/presentationlargespeaker/presentationsmallspeaker/speaker_full>

CustomLayoutName: <S: 1, 128>

Example:

```
xCommand Video PictureLayoutSet Target: Local LayoutFamily: equal
*r SetPictureLayoutResult (status=OK):
** end
```

xCommand Video Layout Add

NOTE: This command is not supported on Codec C40.

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

Requires user role: `USER`

Parameters:

LayoutId: <1..2147483647>

Example:

```
xCommand Video Layout Add
OK
*r VideoLayoutAddResult (status=OK):
 LayoutId: 1
** end
```

xCommand Video Layout Remove

NOTE: This command is not supported on Codec C40.

Removes an existing video layout.

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

Requires user role: `USER`

Parameters:

LayoutId(r): <1..2147483647>

Example:

```
xCommand Video Layout Remove LayoutId: 1
*r VideoLayoutRemoveResult (status=OK):
** end
```

xCommand Video Layout RemoveAll

NOTE: This command is not supported on Codec C40.

Removes all the existing video layouts.

Requires user role: `USER`

Example:

```
xCommand Video Layout RemoveAll
*r VideoLayoutRemoveAllResult (status=OK):
** end
```

xCommand Video Layout Reset

NOTE: This command is not supported on Codec C40.

Resets all the layout compositions to factory default.

Requires user role: `USER`

Example:

```
xCommand Video Layout Reset
*r VideoLayoutResetResult (status=OK):
** end
```

xCommand Video Layout Frame Add

NOTE: This command is not supported on Codec C40.

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

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

FrameId: A unique identifier of the frame.

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

PositionY: The top-left Y position of the frame.

Width: The width of the frame.

Height: The height of the frame.

Layer: The stacking order of the frames (1-5). Must be unique per frame per layout. Layer 5 will be on top.

Border: Select whether or not the frame should have a border.

VideoSourceType: Select the video source to be used as the content of the frame. The parameter arguments are explained below:

Graphic: Not in use.

LocalInput: Select one of the local input sources.

LocalMain: Select what is currently the main source on the local side.

LocalPresentation: Select what is currently the default presentation source on the local side.

MostSpeaking: Select the current loudest speaker in the conference.

OtherMain: Select another remote site in a call. This involves logic in the codec to ensure that a site does not see itself. You can add several frames with OtherMain. The source and the layout engine will automatically populate the frame content with one of the other sites main video, making sure that you do not see the same site more than once, nor yourself.

OwnMain: Select that each site can see its own selfview.

Presentation: Select what is currently the presentation source (there is always only one active H.239 content in a conference).

RemoteMain: Select the main source of remote site.The CallId must then be specified in the VideoSourceId parameter.

RemotePresentation: Select the presentation source of remote site. The CallId must then be specified in the VideoSourceId parameter.

VideoFile: Not in use.

VideoSourceId: A unique identifier of the video source.

If the VideoSourceType is set to RemoteMain the VideoSourceId is the CallId of the remote site.

If the VideoSourceType is set to LocalInput the VideoSourceId is the video input SourceId.

Otherwise set to 1.

Requires user role: `ADMIN`

Parameters:

LayoutId(r): <1..2147483647>

FrameId: <1..65535>

PositionX(r): <0..10000>

PositionY(r): <0..10000>

Width(r): <1..10000>

Height(r): <1..10000>

Layer(r): <1..5>

Border: <On/Off>

VideoSourceType(r): <graphic/localInput/localMain/localPresentation/mostSpeaking/otherMain/ownMain/presentation/remoteMain/remotePresentation/videoFile>

VideoSourceId: <0..2147483647>

Example:

```
xCommand Video Layout Frame Add LayoutId:1 PositionX:100 PositionY:100
Width:9800 Height:9800 Layer:1 Border:off VideoSourceType:localInput
VideoSourceId:1
OK
*r VideoLayoutFrameAddResult (status=OK):
 FrameId: 1
** end
```

xCommand Video Layout Frame Remove

NOTE: This command is not supported on Codec C40.

Remove a video frame from an existing layout.

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

FrameId: A unique identifier of the frame, which was created when the xCommand Video Frame Add command was issued.

Requires user role: `ADMIN`

Parameters:

LayoutId(r): <1..2147483647>

FrameId(r): <1..65535>

Example:

```
xCommand Video Layout Frame Remove LayoutId:1 FrameId:1
*r VideoLayoutFrameRemoveResult (status=OK):
** end
```

xCommand Video Layout Frame Update

NOTE: This command is not supported on Codec C40.

Change the details of an existing frame in a video layout.

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

FrameId: A unique identifier of the frame, which was created when the xCommand Video Frame Add command was issued.

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

PositionY: The top-left Y position of the frame.

Width: The width of the frame.

Height: The height of the frame.

Layer: The stacking order of the frames (1-5). Must be unique per frame per layout. Layer 5 will be on top.

Border: Select whether or not the frame should have a border.

VideoSourceType: Select the video source to be used as the content of the frame. The parameter arguments are explained below:

Graphic: Not in use.

LocalInput: Select one of the local input sources.

LocalMain: Select what is currently the main source on the local side.

LocalPresentation: Select what is currently the default presentation source on the local side.

MostSpeaking: Select the current loudest speaker in the conference.

OtherMain: Select another remote site in a call. This involves logic in the codec to ensure that a site does not see itself. You can add several frames with OtherMain. The source and the layout engine will automatically populate the frame content with one of the other sites main video, making sure that you do not see the same site more than once, nor yourself.

OwnMain: Select that each site can see its own selfview.

Presentation: Select what is currently the presentation source (there is always only one active H.239 content in a conference).

RemoteMain: Select the main source of remote site.The CallId must then be specified in the VideoSourceId parameter.

RemotePresentation: Select the presentation source of remote site. The CallId must then be specified in the VideoSourceId parameter.

VideoFile: Not in use.

VideoSourceId: A unique identifier of the video source.

If the VideoSourceType is set to RemoteMain the VideoSourceId is the CallId of the remote site.

If the VideoSourceType is set to LocalInput the VideoSourceId is the video input SourceId.

Otherwise set to 1.

Requires user role: `USER`

Parameters:

LayoutId(r): <1..2147483647>

FrameId(r): <1..65535>

PositionX: <0..10000>

PositionY: <0..10000>

Width: <1..10000>

Height: <1..10000>

Layer: <1..5>

Border: <On/Off>

VideoSourceType: <graphic/localInput/localMain/localPresentation/mostSpeaking/otherMain/ownMain/presentation/remoteMain/remotePresentation/videoFile>

VideoSourceId: <0..2147483647>

Example:

```
xCommand Video Layout Frame Update LayoutId:1 FrameId:1
PositionX:200 PositionY:200 Width:9600 Height:9600 Layer:1 Border:off
VideoSourceType:localInput VideoSourceId:1
OK
*r VideoLayoutFrameUpdateResult (status=OK):
** end
```

xCommand Video Layout Assign

NOTE: This command is not supported on Codec C40.

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

CallId: A unique identifier for the call. The CallID is returned when the xCommand Dial command is run. During the call you can run the xStatus Call command to see the CallId.

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

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

Requires user role: `USER`

Parameters:

CallId(r): <0..65534>

OutputId(r): <0..65534>

LayoutId(r): <1..2147483647>

Example:

```
xCommand Video Layout Assign CallId:1 OutputId:1 LayoutId:2
OK
*r VideoLayoutAssignResult (status=OK):
** end
```

xCommand Video Layout AssignLocalOutput

NOTE: This command is not supported on Codec C40.

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

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

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

Requires user role: `USER`

Parameters:

LayoutId(r): <1..2147483647>

OutputId(r): <0..65534>

Example:

```
xCommand Video Layout AssignLocalOutput OutputId:1 LayoutId:2
OK
*r VideoLayoutAssignLocalOutputResult (status=OK):
** end
```

xCommand Video Layout AssignCall

NOTE: This command is not supported on Codec C40.

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

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

CallId: A unique identifier for the call. The CallID is returned when the xCommand Dial command is run. During the call you can run the xStatus Call command to see the CallId.

Requires user role: `USER`

Parameters:

LayoutId(r): <1..2147483647>

CallId: <0..65534>

Example:

```
xCommand Video Layout AssignCall LayoutId:2 CallId:1
OK
*r VideoLayoutAssignCallResult (status=OK):
** end
```

xCommand Video Layout AssignPresentation

NOTE: This command is not supported on Codec C40.

Assign the presentation layout (dual stream) to the remote output. The defined video composition will appear on the remote dual video stream. This layout must be full-screen and contain only one frame.

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

Requires user role: `USER`

Parameters:

LayoutId(r): <1..2147483647>

Example:

```
xCommand Video Layout AssignPresentation LayoutId:2
OK
*r VideoLayoutAssignPresentationResult (status=OK):
** end
```

xCommand Video Layout UnAssign

NOTE: This command is not supported on Codec C40.

Remove the defined video layout, and go back to default.

CallId: A unique identifier for the call. The CallID is returned when the xCommand Dial command is run. During the call you can run the xStatus Call command to see the CallId.

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

Requires user role: `USER`

Parameters:

CallId(r): <0..65534>

OutputId(r): <0..65534>

Example:

```
xCommand Video Layout UnAssign CallId: 1 OutputId: 1
*r VideoLayoutUnassignResult (status=OK):
** end
```

xCommand Video Layout UnAssignLocalOutput

NOTE: This command is not supported on Codec C40.

Remove the defined video layout, and go back to default.

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

Requires user role: `USER`

Parameters:

OutputId(r): <0..65534>

Example:

```
xCommand Video Layout UnAssignLocalOutput OutputId:1
OK
*r VideoLayoutUnAssignLocalOutputResult (status=OK):
** end
```

xCommand Video Layout UnAssignCall

NOTE: This command is not supported on Codec C40.

Remove the defined video layout, and go back to default.

CallId: A unique identifier for the call. The CallID is returned when the xCommand Dial command is run. During the call you can run the xStatus Call command to see the CallId.

Requires user role: `USER`

Parameters:

CallId: <0..65534>

Example:

```
xCommand Video Layout UnAssignCall CallId:1
OK
*r VideoLayoutUnAssignCallResult (status=OK):
** end
```

xCommand Video Layout UnAssignPresentation

NOTE: This command is not supported on Codec C40.

Remove the defined video layout, and go back to default.

Requires user role: `USER`

Example:

```
xCommand Video Layout UnAssignPresentation
OK
*r VideoLayoutUnAssignPresentationResult (status=OK):
** end
```

xCommand Video Layout SetPresentationView

Set the presentation view mode.

View: Select Default when you want the presentation to be viewed with the default settings for the codec. Select Maximized when you want the presentation to be displayed in full screen. Select Minimized when you want the presentation to be displayed in a small picture on screen.

Requires user role: `ADMIN`

Parameters:

View(r): <Default/Maximized/Minimized>

Example:

```
xCommand Video Layout SetPresentationView View:Default
OK
*r VideoLayoutSetPresentationViewResult (status=OK):
** end
```

xCommand Video Layout LoadDb

Loads and starts using the specified video layout database. The default video layout database is initially provided by the system. The custom database is generated by the Cisco TC Console tool and is made available to the codec from within the TC Console tool. The TC Console software is found at the Developer Zone web page. Go to: http://developer.tandberg.com/web/guest/tools/integrators/audio-console.

Custom: The system will use the custom video layout database which generated by the Cisco TC Console tool.

CustomAutoModes: The system will use the auto mode part of the custom video layout database.

Default: The system use the default video layout database which is provided by the system.

Requires user role: `USER`

Parameters:

Type(r): <Custom/CustomAutoModes/Default>

Example:

```
xCommand Video Layout LoadDb Type: Default
*r VideoLayoutLoadDbResult (status=OK):
** end
```

The Experimental commands

The Experimental commands are for testing only and should not be used unless agreed with Cisco. These commands are not documented and WILL change in later releases.

Chapter 5

Description of the xStatus commands

Description of the xStatus commands

The following pages will list an example of the xStatus commands and the response. Status type commands returns 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/telepresence/docs

The 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.

Example:

```
xStatus Audio
*s Audio Microphones Mute: Off
*s Audio Volume: 60
*s Audio Input LocalInput 1 Name: "Microphone"
*s Audio Input LocalInput 1 MixerMode: "GainShared"
*s Audio Input LocalInput 1 Mute: "Off"
*s Audio Input LocalInput 1 Channels: 1
*s Audio Input LocalInput 1 AGC: "On"
*s Audio Input LocalInput 1 Connector: "Microphone.1"
*s Audio Input LocalInput 1 Connector: "Microphone.2"
*s Audio Input LocalInput 1 Connector: "Microphone.3"
*s Audio Input LocalInput 1 Connector: "Microphone.4"
*s Audio Input LocalInput 2 Name: "PC input"
*s Audio Input LocalInput 2 MixerMode: "Fixed"
*s Audio Input LocalInput 2 Mute: "Off"
*s Audio Input LocalInput 2 Channels: 2
*s Audio Input LocalInput 2 AGC: "Off"
*s Audio Input LocalInput 2 Connector: "Line.1"
*s Audio Input LocalInput 2 Connector: "Line.2"
*s Audio Input LocalInput 3 Name: "HDMI input"
*s Audio Input LocalInput 3 MixerMode: "Fixed"
*s Audio Input LocalInput 3 Mute: "Off"
*s Audio Input LocalInput 3 Channels: 2
*s Audio Input LocalInput 3 AGC: "Off"
*s Audio Input LocalInput 3 Connector: "HDMI.2"
*s Audio Input RemoteInput 8 CallId: 3
*s Audio Output LocalOutput 4 Name: "Loudspeaker"
*s Audio Output LocalOutput 4 Loudspeaker: "On"
*s Audio Output LocalOutput 4 Channels: 2
*s Audio Output LocalOutput 4 Connector: "Line.1"
*s Audio Output LocalOutput 4 Connector: "Line.2"
*s Audio Output LocalOutput 4 Connector: "HDMI.1"
*s Audio Output LocalOutput 4 Input 2 Gain: 0
*s Audio Output LocalOutput 4 Input 3 Gain: 0
*s Audio Output LocalOutput 5 Name: "Recorder"
*s Audio Output LocalOutput 5 Loudspeaker: "Off"
*s Audio Output LocalOutput 5 Channels: 2
*s Audio Output LocalOutput 5 Input 1 Gain: 0
*s Audio Output RemoteOutput 9 CallId: 3
*s Audio Output RemoteOutput 9 Input 1 Gain: 0
*s Audio Output RemoteOutput 9 Input 2 Gain: 0
*s Audio Output RemoteOutput 9 Input 3 Gain: 0
*s Audio Module 0 Type: Unknown
*s Audio Module 0 SoftwareID: ""
*s Audio Module 0 HardwareID: ""
*s Audio Module 0 Connector: ""
** end
```

xStatus Audio Microphones Mute

Shows the microphones mute mode.

Value space of the result returned:

<On/Off>

Example:

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

xStatus Audio Volume

Shows the volume level (dB) of the loudspeaker output.

Value space of the result returned:

<0..100>

Example:

```
xStatus Audio Volume
*s Audio Volume: 70
** end
```

xStatus Audio Input LocalInput [1..n] Name

Shows the name of the local input. A local input is a mix of input connectors. You can run the command xStatus Audio Input LocalInput to find the identity [1..n] of the input.

Value space of the result returned:

<String>

Example:

```
xStatus Audio Input LocalInput 1 Name
*s Audio Input LocalInput 1 Name: "Microphone"
** end
```

xStatus Audio Input LocalInput [1..n] MixerMode

Shows how the local input connectors have been mixed together. You can run the command xStatus Audio Input LocalInput to find the identity [1..n] of the input.

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

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

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

Value space of the result returned:

<"Auto"/"Fixed"/"GainShared">

Example:

```
xStatus Audio Input LocalInput 1 MixerMode
*s Audio Input LocalInput 1 MixerMode: "Auto"
** end
```

xStatus Audio Input LocalInput [1..n] Mute

Shows the mute mode for the local inputs. A local input is a mix of input connectors.

You can run the command xStatus Audio Input LocalInput to find the identity [1..n] of the input.

Value space of the result returned:

<"On"/"Off">

Example:

```
xStatus Audio Input LocalInput 1 Mute
*s Audio Input LocalInput 1 Mute: "Off"
** end
```

xStatus Audio Input LocalInput [1..n] Channels

States if the local input channels are mixed into into a mono signal (1) or stereo signal (2).

You can run the command xStatus Audio Input LocalInput to find the identity [1..n] of the input.

Value space of the result returned:

<1..2>

Example:

```
xStatus Audio Input LocalInput 1 Channels
*s Audio Input LocalInput 1 Channels: 1
** end
```

xStatus Audio Input LocalInput [1..n] AGC

Shows the AGC (Automatic Gain Control) mode on the local input.

You can run the command xStatus Audio Input LocalInput to find the identity [1..n] of the input.

Value space of the result returned:

<"On"/"Off">

Example:

```
xStatus Audio Input LocalInput 1 AGC
*s Audio Input LocalInput 1 AGC: "On"
** end
```

xStatus Audio Input LocalInput [1..n] Connector

Lists the connectors that are attached to the local input.

You can run the command xStatus Audio Input LocalInput to find the identity [1..n] of the input.

Value space of the result returned:

<"Microphone.1"/../"Microphone.4"/"Line.1"/"Line.2"/"HDMI.2">

Example:

```
xStatus Audio Input LocalInput 1 Connector
*s Audio Input LocalInput 1 Connector: "Microphone.1"
** end
```

xStatus Audio Input RemoteInput [1..n] CallId

Shows the CallId for the remote audio input.

You can run the command xStatus Audio Input RemoteInput to find the identity [1..n] of the input.

Value space of the result returned:

<0..65534>

Example:

```
xStatus Audio Input RemoteInput 8 CallId
*s Audio Input RemoteInput 8 CallId: 28
** end
```

xStatus Audio Output LocalOutput [1..n] Name

Shows the name of the local output.

You can run the command xStatus Audio Output LocalOutput to find the identity [1..n] of the output.

Value space of the result returned:

<String>

Example:

```
xStatus Audio Output LocalOutput 4 Name
*s Audio Output LocalOutput 4 Name: "MyLocalOutput1"
** end
```

xStatus Audio Output LocalOutput [1..n] Loudspeaker

Shows the Loudspeaker mode on the local output. If one or more of the output connectors that are attached to the local output are connected to a loudspeaker, then this signal should be a reference signal to the echo canceller and Loudspeaker should be set to On.

You can run the command xStatus Audio Output LocalOutput to find the identity [1..n] of the output.

Value space of the result returned:

<"On"/"Off">

Example:

```
xStatus Audio Output LocalOutput 4 Loudspeaker
*s Audio Output LocalOutput 4 Loudspeaker: "Off"
** end
```

xStatus Audio Output LocalOutput [1..n] Channels

States if the local output channels are mixed into into a mono signal (1) or stereo signal (2).

You can run the command xStatus Audio Output LocalOutput to find the identity [1..n] of the output.

Value space of the result returned:

<1..2>

Example:

```
xStatus Audio Output LocalOutput 4 Channels
*s Audio Output LocalOutput 4 Channels: "1"
** end
```

xStatus Audio Output LocalOutput [1..n] Connector

Lists the connectors that are attached to the local output.

You can run the command xStatus Audio Output LocalOutput to find the identity [1..n] of the output.

Value space of the result returned:

<"Line.1"/"Line.2"/"HDMI.1"/"HDMI.2">

Example:

```
xStatus Audio Output LocalOutput 4 Connector
*s Audio Output LocalOutput 4 Connector: "Line.1"
** end
```

xStatus Audio Output LocalOutput [1..n] Input [1..n] Gain

Shows the gain (dB) on the input, when it is connected to the local output. Range from -54 dB to 15 dB, where -54 dB equals Off.

You can run the command xStatus Audio Output LocalOutput to find the identity [1..n] of the output and input.

Value space of the result returned:

<-54..15>

Example:

```
xStatus Audio Output LocalOutput 4 Input 2 Gain
*s Audio Output LocalOutput 4 Input 2 Gain: 0
** end
```

xStatus Audio Output RemoteOutput [1..n] CallId

Shows the CallId for the remote audio output.

You can run the command xStatus Audio Output RemoteOutput to find the identity [1..n] of the output.

Value space of the result returned:

<1..65534>

Example:

```
xStatus Audio Output RemoteOutput 9 CallId
*s Audio Output RemoteOutput 9 CallId: 28
** end
```

xStatus Audio Output RemoteOutput [1..n] Input [1..n] Gain

Shows the gain (dB) on the input, when input is connected to the remote output. Range from -54 dB to 15 dB, where -54 dB equals Off.

You can run the command xStatus Audio Output RemoteOutput to find the identity [1..n] of the output and input.

Value space of the result returned:

<-54..15>

Example:

```
xStatus Audio Output RemoteOutput 9 Input 1 Gain
*s Audio Output RemoteOutput 9 Input 1 Gain: 0
** end
```

xStatus Audio Module [0] Type

Shows the audio module type. If the module type is DigitalNAM (Digital Natural Audio Module) you can also read the SoftwareId and HardwareId.

Value space of the result returned:

<DigitalNAM/Unknown>

Example:

```
xStatus Audio Module 0 Type
*s Audio Module 0 Type: DigitalNAM
** end
```

xStatus Audio Module [0] SoftwareID

Shows the SoftwareID of the DNAM dsp software.

Value space of the result returned:

<String>

Example:

```
xStatus Audio Module 0 SoftwareID
*s Audio Module 0 SoftwareID: "114"
** end
```

xStatus Audio Module [0] HardwareID

Shows the DNAM HardwareID.

Value space of the result returned:

<String>

Example:

```
xStatus Audio Module 0 HardwareID
*s Audio Module 0 HardwareID: "B40F69"
** end
```

xStatus Audio Module [0] Connector

Shows which audio output connector the audio module is attached to.

Value space of the result returned:

<"Line_out.1">

Example:

```
xStatus Audio Module 0 Connector
*s Audio Module 0 Connector: "Line _ out.1"
** end
```

The Call status

xStatus Call

Shows the top level overview of the call status. The call identity is used when query for additional information about the call.

Example:

```
xStatus Call
*s Call 27 Status: Connected
*s Call 27 Direction: Outgoing
*s Call 27 Protocol: "sip"
*s Call 27 CallType: Video
*s Call 27 RemoteNumber: "firstname.lastname@company.com"
*s Call 27 CallbackNumber: "sip:firstname.lastname@company.com"
*s Call 27 DisplayName: "Firstname Lastname"
*s Call 27 TransmitCallRate: 3968
*s Call 27 ReceiveCallRate: 4000
*s Call 27 FacilityServiceId: 0
*s Call 27 Encryption Type: "None"
*s Call 27 PlacedOnHold: False
*s Call 27 Duration: 2354
** end
```

xStatus Call [1..n] Status

Shows the status of a call. You can run the command xStatus Call to find the call identity.

Value space of the result returned:

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

Example:

```
xStatus Call 27 Status
*s Call 27 Status: Connected
** end
```

xStatus Call [1..n] Direction

States the direction of the call initiation. You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<Incoming/Outgoing>

Example:

```
xStatus Call 27 Direction
*s Call 27 Direction: Outgoing
** end
```

xStatus Call [1..n] Protocol

Shows the call protocol of the incoming or outgoing call. You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<H323/SIP>

Example:

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

xStatus Call [1..n] CallType

Shows the call type of the incoming or outgoing call. You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<Video/Audio>

Example:

```
xStatus Call 27 CallType
*s Call 27 CallType: Video
** end
```

xStatus Call [1..n] RemoteNumber

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

Value space of the result returned:

<String>

Example:

```
xStatus Call 27 RemoteNumber
*s Call 27 RemoteNumber: "5585232"
** end
```

xStatus Call [1..n] CallbackNumber

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

Value space of the result returned:

<String>

Example:

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

xStatus Call [1..n] DisplayName

Shows the name of the remote (far end) participant in an incoming or outgoing call. You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<String>

Example:

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

xStatus Call [1..n] TransmitCallRate

Shows the transmit bandwidth in the call in kilobits per second (kbps). You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<Integer>

Example:

```
xStatus Call 27 TransmitCallRate
*s Call 27 TransmitCallRate: 768
** end
```

xStatus Call [1..n] ReceiveCallRate

Shows the receive bandwidth in the call in kilobits per second (kbps). You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<Integer>

Example:

```
xStatus Call 27 ReceiveCallRate
*s Call 27 ReceiveCallRate: 4000
** end
```

xStatus Call [1..n] FacilityServiceId

When calling a facility service, the facility service id is shown. Otherwise the value 0 is returned.

Value space of the result returned:

<0..5>

Example:

```
xStatus Call FacilityServiceId
*s Call 3 FacilityServiceId: 1
** end
```

xStatus Call [1..n] Encryption Type

Shows the encryption type of the call. You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<"None"/"Aes-128">

Example:

```
xStatus Call 27 Encryption Type
*s Call 27 Encryption Type: "None"
** end
```

xStatus Call [1..n] PlacedOnHold

Shows the placed on hold status of the call. You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<True/False>

Example:

```
xStatus Call 27 PlacedOnHold
*s Call 27 PlacedOnHold: False
** end
```

xStatus Call [1..n] Duration

Shows the duration of a call (in seconds). You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<Integer>

Example:

```
xStatus Call 27 Duration
*s Call 27 Duration: 2354
** end
```

The Camera status

xStatus Camera

Shows the top level overview of the camera status.

Example:

```
xStatus Camera
*s Camera 1 Connected: True
*s Camera 1 HardwareID: "50000000"
*s Camera 1 Manufacturer: "TANDBERG"
*s Camera 1 Model: "PrecisionHD 1080p 12X"
*s Camera 1 SoftwareID: "S01718-4.0FINAL [ID:40063] 2010-10-20"
*s Camera 1 SerialNumber: "B1AB26B00010"
*s Camera 1 IpAddress: ""
*s Camera 1 MacAddress: ""
*s Camera 1 Position Pan: 412
*s Camera 1 Position Tilt: 106
*s Camera 1 Position Zoom: 828
*s Camera 1 Position Focus: 4597
*s Camera 1 Capabilities Options: "ptzf"
*s Camera 1 Flip: "Off"
 -
 - //the list is continued with Camera 2-7//
** end
```

xStatus Camera [1..7] Connected

Shows if the camera is connected or not.

Value space of the result returned:

<True/False>

Example:

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

xStatus Camera [1..7] HardwareID

Shows the hardware identity of the camera.

Value space of the result returned:

<String>

Example:

```
xStatus Camera 1 HardwareID
*s Camera 1 HardwareID: "50000000"
** end
```

xStatus Camera [1..7] Manufacturer

Shows the manufacturer of the camera.

Value space of the result returned:

<String>

Example:

```
xStatus Camera 1 Manufacturer
*s Camera 1 Manufacturer: "TANDBERG"
** end
```

xStatus Camera [1..7] Model

Shows the camera model.

Value space of the result returned:

<String>

Example:

```
xStatus Camera 1 Model
*s Camera 1 Model: "PrecisionHD 1080p 12X"
** end
```

xStatus Camera [1..7] SoftwareID

Shows the software identity of the camera.

Value space of the result returned:

<String>

Example:

```
xStatus Camera 1 SoftwareID
*s Camera 1 SoftwareID: "S01718-4.0FINAL [ID:40063] 2010-10-20"
** end
```

xStatus Camera [1..7] SerialNumber

Shows the camera serial number.

Value space of the result returned:

<String>

Example:

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

xStatus Camera [1..7] IpAddress

Shows the camera IP address.

Value space of the result returned:

<String>

Example:

```
xStatus Camera 1 IpAddress
*s Camera 1 IpAddress: ""
** end
```

xStatus Camera [1..7] MacAddress

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

Value space of the result returned:

<String>

Example:

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

xStatus Camera [1..7] Position Pan

Shows the current pan (move left and right) position of the camera. The value range depends on camera type.

Value space of the result returned:

<-65535..65535>

Example:

```
xStatus Camera 1 Position Pan
*s Camera 1 Position Pan: 412
** end
```

xStatus Camera [1..7] Position Tilt

Shows the current tilt (move up and down) position of the camera. The value range depends on camera type.

Value space of the result returned:

<-65535..65535>

Example:

```
xStatus Camera 1 Position Tilt
*s Camera 1 Position Tilt: 106
** end
```

xStatus Camera [1..7] Position Zoom

Shows the current zoom (zoom in and out) position of the camera. The value range depends on camera type.

Value space of the result returned:

<0..65535>

Example:

```
xStatus Camera 1 Position Zoom
*s Camera 1 Position Zoom: 828
** end
```

xStatus Camera [1..7] Position Focus

Shows the current focus position of the camera. The value range depends on camera type.

Value space of the result returned:

<0..65535>

Example:

```
xStatus Camera 1 Position Focus
*s Camera 1 Position Focus: 4597
** end
```

xStatus Camera [1..7] Capabilities Options

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

Value space of the result returned:

<String>

Example:

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

xStatus Camera [1..7] Flip

In Flip mode (vertical flip) the image can be flipped upside down.

Value space of the result returned:

<"Auto"/"On"/"Off">

Example:

```
xStatus Camera 1 Flip
*s Camera 1 Flip: "Off"
** end
```

The Conference status

xStatus Conference

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

Example:

```
xStatus Conference
*s Conference Presentation Mode: Off
*s Conference Presentation Protocol: ""
*s Conference Presentation Resolution Height: 0
*s Conference Presentation Resolution Width: 0
*s Conference Presentation SiteId: 0
*s Conference Presentation LocalSource: 0
*s Conference Multipoint Mode: "MultiWay"
*s Conference DoNotDisturb: Inactive
*s Conference Site 26 MicrophonesMuted: True
*s Conference Site 26 Capabilities FECC NumberOfPresets: 15
*s Conference Site 26 Capabilities FECC NumberOfSources: 5
*s Conference Site 26 Capabilities FECC Source 1 SourceId: 6
*s Conference Site 26 Capabilities FECC Source 1 Name: "Main camera"
*s Conference Site 26 Capabilities FECC Source 1 Options: "ptzf"
*s Conference Site 26 Capabilities FECC Source 2 SourceId: 7
*s Conference Site 26 Capabilities FECC Source 2 Name: "PC"
*s Conference Site 26 Capabilities FECC Source 2 Options: ""
*s Conference Site 26 Capabilities FECC Mode: On
*s Conference Site 26 Manufacturer: "Cisco"
*s Conference Site 26 SoftwareID: "TC5"
*s Conference Site 26 BlackScreenCause: None
*s Conference Site 26 ConferenceExtended: NotExtended
*s Conference Site 26 BookingId: "MyConference"
** end
```

xStatus Conference Presentation Mode

Shows the status of the secondary video stream.

Value space of the result returned:

<Off/Sending/Receiving>

Example:

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

xStatus Conference Presentation Protocol

Shows the video protocol used when transmitting the presentation.

Value space of the result returned:

<String>

Example:

```
xStatus Conference Presentation Protocol
*s Conference Presentation Protocol: "H264"
** end
```

xStatus Conference Presentation Resolution Height

Shows the height of the the presentation.

Value space of the result returned:

<0..3000>

Example:

```
xStatus Conference Presentation Resolution Height
*s Conference Presentation Resolution Height: 0
** end
```

xStatus Conference Presentation Resolution Width

Shows the width of the the presentation.

Value space of the result returned:

<0..4000>

Example:

```
xStatus Conference Presentation Resolution Width
*s Conference Presentation Resolution Width: 0
** end
```

xStatus Conference Presentation SiteId

Shows the identity of the system that sends the presentation.

Value space of the result returned:

<0..65535>

Example:

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

xStatus Conference Presentation LocalSource

Shows the local video input source that is used when the presentation is sent from the local site.

Value space of the result returned:

<1..5>

Example:

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

xStatus Conference Site [1..n] Capabilities FECC NumberOfPresets

Shows the number of presets available for the input sources at a far end site.

Value space of the result returned:

<1..15>

Example:

```
xStatus Conference Site 2 Capabilities FECC NumberOfPresets
*s Conference Site 2 Capabilities FECC NumberOfPresets: 15
** end
```

xStatus Conference Site [1..n] Capabilities FECC NumberOfSources

Shows the number of input sources that can be connected at a far end site.

Value space of the result returned:

<1..5>

Example:

```
xStatus Conference Site 2 Capabilities FECC NumberOfSources
*s Conference Site 2 Capabilities FECC NumberOfSources: 5
** end
```

xStatus Conference Site [1..n] Capabilities FECC Source [1..n] SourceId

Shows the ID of an input source that can be connected at a far end site.

Value space of the result returned:

<Integer>

Example:

```
xStatus Conference Site 2 Capabilities FECC Source 1 SourceId
*s Conference Site 2 Capabilities FECC Source 1 SourceId: 6
** end
```

xStatus Conference Site [1..n] Capabilities FECC Source [1..n] Name

Shows the name of an input source that can be connected at a far end site.

Value space of the result returned:

<String>

Example:

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

xStatus Conference Site [1..n] Capabilities FECC Source [1..n] Options

Shows available options for an input source that can be connected at a far end site (for a camera: p=pan; t=tilt; z=zoom; f=focus).

Value space of the result returned:

<String>

Example:

```
xStatus Conference Site 2 Capabilities FECC Source 1 Options
*s Conference Site 2 Capabilities FECC Source 1 Options: "ptzf"
** end
```

xStatus Conference Site [1..n] Capabilities FECC Mode

Shows whether or not you have permission to control the input sources at a far end site.

On: Far end input source control is permitted.

Off: Far end input source control is not permitted.

Value space of the result returned:

<On/Off>

Example:

```
xStatus Conference Site 2 Capabilities FECC Mode
*s Conference Site 2 Capabilities FECC Mode: On
** end
```

xStatus Conference Site [1..n] Capabilities Presentation

Lists the presentation capabilities for other participants in the conference.

Value space of the result returned:

<True/False>

Example:

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

xStatus Conference Site [1..n] MicrophonesMuted

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

Value space of the result returned:

<True/False>

Example:

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

xStatus Conference Site [1..n] Manufacturer

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

Value space of the result returned:

<String>

Example:

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

xStatus Conference Site [1..n] SoftwareID

Shows the ID of the software running of the video system at a far end site.

Value space of the result returned:

<String>

Example:

```
xStatus Conference Site 2 SoftwareID
*s Conference Site 2 SoftwareID: "TC5"
** end
```

xStatus Conference Site [1..n] BlackScreenCause

Shows the reason why the screen is black while in a CTMS (Cisco TelePresence Multipoint Switch) managed conference.

FirstParticipant: You are the first and only participant in the meeting.

LastParticipant: You are the only remaining participant in the meeting.

NoResources: The conference started with insufficient resources.

EarlyJoin: The scheduled conference hast not started yet.

HostNotJoined: The conference is waiting for the host to join.

SecurityIssues: The video is blocked due to security issues.

NoVideo: It is a conference without video.

WebexOnly: It is a Webex only conference.

Value space of the result returned:

<None/Other/FirstParticipant/LastParticipant/NoResources/EarlyJoin/HostNotJoined/SecurityIssues/NoVideo/WebexOnly>

Example:

```
xStatus Conference Site 2 BlackScreenCause
*s Conference Site 17 BlackScreenCause: None
** end
```

xStatus Conference Site [1..n] ConferenceExtended

Shows the status of conference extension (only relevant for conferences scheduled from CTS-MAN (Cisco TelePresence Manager); they can be extended only once).

Unsupported: Conference extension is not supported.

NotExtended: The conference is not (yet) extended.

Extended: The conference has been extended, and is currently in the extended phase.

Value space of the result returned:

<Unsupported/NotExtended/Extended>

Example:

```
xStatus Conference Site 2 ConferenceExtended
*s Conference Site 2 ConferenceExtended: NotExtended
** end
```

xStatus Conference Site [1..n] BookingId

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

Value space of the result returned:

<String>

Example:

```
xStatus Conference Site 2 BookingId
*s Conference Site 33 BookingId: "MyConference"
** end
```

xStatus Conference Multipoint Mode

Shows how multipoint video conferences are handled. See xConfiguration Conference Multipoint Mode for more information.

Value space of the result returned:

<"MultiSite"/"MultiWay"/"Off">

Example:

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

xStatus Conference DoNotDisturb

Shows whether DoNotDisturb mode is switched on or not.

Value space of the result returned:

<Active/Inactive>

Example:

```
xStatus Conference DoNotDisturb
*s Conference DoNotDisturb: Inactive
** end
```

The 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.

Example:

```
xStatus Diagnostics
*s Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 Jitter: 0
*s Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 Packets: 132505
*s Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 Loss: 0
*s Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 LastIntervalLost: 0
*s Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 LastIntervalReceived: 84
*s Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 Drop: 0
*s Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 Bytes: 21200960
*s Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 ChannelRate: 64000
*s Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 MaxJitter: 0
*s Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 Jitter: 6
*s Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 Packets: 133166
*s Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 Loss: 0
*s Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 LastIntervalLost: 0
*s Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 LastIntervalReceived: 148
*s Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 Drop: 0
*s Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 Bytes: 122301901
*s Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 ChannelRate: 506000
*s Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 MaxJitter: 9
*s Diagnostics Call 27 Channels IncomingDataChannel 335 Netstat 1 Jitter: 0
*s Diagnostics Call 27 Channels IncomingDataChannel 335 Netstat 1 Packets: 0
*s Diagnostics Call 27 Channels IncomingDataChannel 335 Netstat 1 Loss: 0
*s Diagnostics Call 27 Channels IncomingDataChannel 335 Netstat 1 LastIntervalLost: 0
*s Diagnostics Call 27 Channels IncomingDataChannel 335 Netstat 1 LastIntervalReceived: 0
*s Diagnostics Call 27 Channels IncomingDataChannel 335 Netstat 1 Drop: 0
*s Diagnostics Call 27 Channels IncomingDataChannel 335 Netstat 1 Bytes: 0
*s Diagnostics Call 27 Channels IncomingDataChannel 335 Netstat 1 ChannelRate: 0
*s Diagnostics Call 27 Channels IncomingDataChannel 335 Netstat 1 MaxJitter: 0
*s Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 Jitter: 0
*s Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 Packets: 0
*s Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 Loss: 0
*s Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 LastIntervalLost: 0
*s Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 LastIntervalReceived: 300
*s Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 Drop: 0
*s Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 Bytes: 0
*s Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 ChannelRate: 0
*s Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 MaxJitter: 0
*s Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 Jitter: 4
*s Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 Packets: 123043
*s Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 Loss: 0
*s Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 LastIntervalLost: 0
*s Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 LastIntervalReceived: 351
*s Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 Drop: 0
*s Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 Bytes: 94720991
*s Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 ChannelRate: 493000
*s Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 MaxJitter: 4
*s Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 Jitter: 0
*s Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 Packets: 0
*s Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 Loss: 0
*s Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 LastIntervalLost: 0
*s Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 LastIntervalReceived: 0
*s Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 Drop: 0
*s Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 Bytes: 0
```

```
*s Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1
ChannelRate: 0
*s Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 MaxJitter:
0
** end
```

xStatus Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1 Jitter

xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 Jitter

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 Jitter

xStatus Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 Jitter

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 Jitter

xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 Jitter

Shows the jitter at the present moment in the incoming/outgoing channel, as specified by RFC 3550.

Value space of the result returned:

<Integer>

Example:

```
xStatus Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1
Jitter
*s Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1 Jitter: 0
** end
```

xStatus Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1 Packets

xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 Packets

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 Packets

xStatus Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 Packets

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 Packets

xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 Packets

Shows the number of packets received/sent in the incoming/outgoing channels.

Value space of the result returned:

<Integer>

Example:

```
xStatus Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1
Packets
*s Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1 Packets:
405
** end
```

xStatus Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1 Loss

xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 Loss

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 Loss

xStatus Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 Loss

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 Loss

xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 Loss

Shows the number of packets lost in the incoming/outgoing channels.

Value space of the result returned:

<Integer>

Example:

```
xStatus Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1 Loss
*s Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1 Loss: 96
** end
```

xStatus Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1 LastIntervalLost

xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 LastIntervalLost

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 LastIntervalLost

xStatus Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 LastIntervalLost

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 LastIntervalLost

xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 LastIntervalLost

Shows the number of packets lost during the last interval for the incoming/outgoing channels.

Value space of the result returned:

<Integer>

Example:

```
xStatus Diagnostics Call 27 Channels IncomingDataChannel 327 Netstat 1
LastIntervalLost
*s Diagnostics Call 27 Channels IncomingDataChannel 327 Netstat 1
LastIntervalLost: 0
** end
```

xStatus Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1 LastIntervalReceived

xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 LastIntervalReceived

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 LastIntervalReceived

xStatus Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 LastIntervalReceived

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 LastIntervalReceived

xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 LastIntervalReceived

Shows the number of packets received during the last interval for the incoming/outgoing channels.

Value space of the result returned:

<Integer>

Example:

```
xStatus Diagnostics Call 27 Channels IncomingDataChannel 327 Netstat 1
LastIntervalReceived
*s Diagnostics Call 27 Channels IncomingDataChannel 327 Netstat 1
LastIntervalReceived: 84
** end
```

xStatus Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1 Drop

xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 Drop

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 Drop

xStatus Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 Drop

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 Drop

xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 Drop

Shows the number of packets dropped in the incoming/outgoing channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1 Drop
*s Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1 Drop: 0
** end
```

xStatus Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1 Bytes

xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 Bytes

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 Bytes

xStatus Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 Bytes

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 Bytes

xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 Bytes

Shows the number of bytes received/sent in the incoming/outgoing channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1 Bytes
*s Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1 Bytes:
129920
** end
```

xStatus Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1 ChannelRate

xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 ChannelRate

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 ChannelRate

xStatus Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 ChannelRate

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 ChannelRate

xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 ChannelRate

Shows the bandwidth for the incoming/outgoing channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1
ChannelRate
*s Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1
ChannelRate: 128000
** end
```

xStatus Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1 MaxJitter

xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 MaxJitter

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 MaxJitter

xStatus Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 MaxJitter

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 MaxJitter

xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 MaxJitter

Shows the maximum jitter that has been measured during last time interval (5 seconds).

Value space of the result returned:

<Integer>

Example:

```
xStatus Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1
MaxJitter
*s Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1 MaxJitter:
0
** end
```

The GPIO status

xStatus GPIO Pin [1..4] State

NOTE: This command is not supported on Codec C40.

Shows the current state of each GPIO pin. The default state is High (+12V). When activated the state is Low (0V).

Value space of the result returned:

<High/Low>

Example:

```
xStatus GPIO Pin 1 State
*s GPIO Pin 1 State: High
** end
```

The H323 status

xStatus H323

Shows the top level overview of the H323 status.

Example:

```
xStatus H323
*s H323 Gatekeeper Status: Registered
*s H323 Gatekeeper Address: "192.0.1.20"
*s H323 Gatekeeper Port: 1719
*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 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
```

The HttpFeedback status

xStatus HttpFeedback

Shows the top level overview of the HTTP status.

Example:

```
xStatus HttpFeedback
*s HttpFeedback 1 URL: "http://tms.group.company.com/tms/public/feedback/
code.aspx"
*s HttpFeedback 1 Expression: "/History/CallLog/History"
*s HttpFeedback 1 Expression: "/Status/Call[Status='connected']"
*s HttpFeedback 1 Expression: "/Status/H323/Gatekeeper"
*s HttpFeedback 1 Expression: "/Status/Ethernet"
*s HttpFeedback 1 Expression: "/Event/CallSuccessful"
*s HttpFeedback 1 Expression: ""
*s HttpFeedback 1 Expression: ""
*s HttpFeedback 1 Expression: ""
*s HttpFeedback 1 Expression: ""
*s HttpFeedback 1 Expression: ""
*s HttpFeedback 1 Expression: ""
*s HttpFeedback 1 Expression: ""
*s HttpFeedback 1 Expression: ""
*s HttpFeedback 1 Expression: ""
*s HttpFeedback 1 Expression: ""
 -
 - continues with HttpFeedback 2-4.
** 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
```

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>

Example:

```
xStatus HttpFeedback 1 Expression
*s HttpFeedback 1 Expression: "/History/CallLog/History"
*s HttpFeedback 1 Expression: "/Status/Call[Status='connected']"
*s HttpFeedback 1 Expression: "/Status/H323/Gatekeeper"
*s HttpFeedback 1 Expression: "/Status/Ethernet"
*s HttpFeedback 1 Expression: "/Event/CallSuccessful"
*s HttpFeedback 1 Expression: ""
*s HttpFeedback 1 Expression: ""
*s HttpFeedback 1 Expression: ""
*s HttpFeedback 1 Expression: ""
*s HttpFeedback 1 Expression: ""
*s HttpFeedback 1 Expression: ""
*s HttpFeedback 1 Expression: ""
*s HttpFeedback 1 Expression: ""
*s HttpFeedback 1 Expression: ""
*s HttpFeedback 1 Expression: ""
** end
```

The 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.

Example:

```
xStatus MediaChannels
*s MediaChannels Call 4 IncomingAudioChannel 41 Encryption Status: Off
*s MediaChannels Call 4 IncomingAudioChannel 41 Audio Protocol: AACLD
*s MediaChannels Call 4 IncomingAudioChannel 41 Audio Mute: False
*s MediaChannels Call 4 IncomingAudioChannel 41 Audio Channels: 1
*s MediaChannels Call 4 IncomingAudioChannel 41 Transport RTP Local
IpAddress: "10.54.86.241"
*s MediaChannels Call 4 IncomingAudioChannel 41 Transport RTP Local Port:
16402
*s MediaChannels Call 4 IncomingAudioChannel 41 Transport RTP Remote
IpAddress: "10.54.86.240"
*s MediaChannels Call 4 IncomingAudioChannel 41 Transport RTP Remote Port:
2334
*s MediaChannels Call 4 IncomingAudioChannel 41 Transport RTCP Local
IpAddress: "10.54.86.241"
*s MediaChannels Call 4 IncomingAudioChannel 41 Transport RTCP Local Port:
16403
*s MediaChannels Call 4 IncomingAudioChannel 41 Transport RTCP Remote
IpAddress: "10.54.86.240"
*s MediaChannels Call 4 IncomingAudioChannel 41 Transport RTCP Remote Port:
2335
*s MediaChannels Call 4 IncomingVideoChannel 44 Encryption Status: Off
*s MediaChannels Call 4 IncomingVideoChannel 44 ChannelRole: Main
*s MediaChannels Call 4 IncomingVideoChannel 44 Video Protocol: H264
*s MediaChannels Call 4 IncomingVideoChannel 44 Video FrameRate: 30
*s MediaChannels Call 4 IncomingVideoChannel 44 Video ResolutionX: 1920
*s MediaChannels Call 4 IncomingVideoChannel 44 Video ResolutionY: 1080
*s MediaChannels Call 4 IncomingVideoChannel 44 Transport RTP Local
IpAddress: "10.54.86.241"
*s MediaChannels Call 4 IncomingVideoChannel 44 Transport RTP Local Port:
16404
*s MediaChannels Call 4 IncomingVideoChannel 44 Transport RTP Remote
IpAddress: "10.54.86.240"
*s MediaChannels Call 4 IncomingVideoChannel 44 Transport RTP Remote Port:
2336
*s MediaChannels Call 4 IncomingVideoChannel 44 Transport RTCP Local
IpAddress: "10.54.86.241"
*s MediaChannels Call 4 IncomingVideoChannel 44 Transport RTCP Local Port:
16405
*s MediaChannels Call 4 IncomingVideoChannel 44 Transport RTCP Remote
IpAddress: "10.54.86.240"
*s MediaChannels Call 4 IncomingVideoChannel 44 Transport RTCP Remote Port:
2337
*s MediaChannels Call 4 IncomingVideoChannel 47 Encryption Status: Off
*s MediaChannels Call 4 IncomingVideoChannel 47 ChannelRole: Presentation
*s MediaChannels Call 4 IncomingVideoChannel 47 Video Protocol: Off
*s MediaChannels Call 4 IncomingVideoChannel 47 Video FrameRate: 0
*s MediaChannels Call 4 IncomingVideoChannel 47 Video ResolutionX: 0
*s MediaChannels Call 4 IncomingVideoChannel 47 Video ResolutionY: 0
*s MediaChannels Call 4 IncomingVideoChannel 47 Transport RTP Local
IpAddress: "10.54.86.241"
*s MediaChannels Call 4 IncomingVideoChannel 47 Transport RTP Local Port:
16406
*s MediaChannels Call 4 IncomingVideoChannel 47 Transport RTP Remote
IpAddress: ""
*s MediaChannels Call 4 IncomingVideoChannel 47 Transport RTP Remote Port: 0
*s MediaChannels Call 4 IncomingVideoChannel 47 Transport RTCP Local
IpAddress: "10.54.86.241"
*s MediaChannels Call 4 IncomingVideoChannel 47 Transport RTCP Local Port:
16407
*s MediaChannels Call 4 IncomingVideoChannel 47 Transport RTCP Remote
IpAddress: ""
*s MediaChannels Call 4 IncomingVideoChannel 47 Transport RTCP Remote Port:
0
*s MediaChannels Call 4 IncomingVideoChannel 51 Encryption Status: Off
*s MediaChannels Call 4 IncomingVideoChannel 51 ChannelRole: Legacy
*s MediaChannels Call 4 IncomingVideoChannel 51 Video Protocol: Off
*s MediaChannels Call 4 IncomingVideoChannel 51 Video FrameRate: 0
*s MediaChannels Call 4 IncomingVideoChannel 51 Video ResolutionX: 0
*s MediaChannels Call 4 IncomingVideoChannel 51 Video ResolutionY: 0
*s MediaChannels Call 4 IncomingVideoChannel 51 Transport RTP Local
IpAddress: "10.54.86.241"
*s MediaChannels Call 4 IncomingVideoChannel 51 Transport RTP Local Port:
16410
*s MediaChannels Call 4 IncomingVideoChannel 51 Transport RTP Remote
IpAddress: ""
*s MediaChannels Call 4 IncomingVideoChannel 51 Transport RTP Remote Port: 0
*s MediaChannels Call 4 IncomingVideoChannel 51 Transport RTCP Local
IpAddress: "10.54.86.241"
*s MediaChannels Call 4 IncomingVideoChannel 51 Transport RTCP Local Port:
16411
```

```
*s MediaChannels Call 4 IncomingVideoChannel 51 Transport RTCP Remote
IpAddress: ""
*s MediaChannels Call 4 IncomingVideoChannel 51 Transport RTCP Remote Port:
0
*s MediaChannels Call 4 OutgoingAudioChannel 42 Encryption Status: Off
*s MediaChannels Call 4 OutgoingAudioChannel 42 Audio Protocol: AACLD
*s MediaChannels Call 4 OutgoingAudioChannel 42 Audio Channels: 1
*s MediaChannels Call 4 OutgoingAudioChannel 42 Transport RTP Local
IpAddress: "10.54.86.241"
*s MediaChannels Call 4 OutgoingAudioChannel 42 Transport RTP Local Port:
16402
*s MediaChannels Call 4 OutgoingAudioChannel 42 Transport RTP Remote
IpAddress: "10.54.86.240"
*s MediaChannels Call 4 OutgoingAudioChannel 42 Transport RTP Remote Port:
2334
*s MediaChannels Call 4 OutgoingAudioChannel 42 Transport RTCP Local
IpAddress: "10.54.86.241"
*s MediaChannels Call 4 OutgoingAudioChannel 42 Transport RTCP Local Port:
16403
*s MediaChannels Call 4 OutgoingAudioChannel 42 Transport RTCP Remote
IpAddress: "10.54.86.240"
*s MediaChannels Call 4 OutgoingAudioChannel 42 Transport RTCP Remote Port:
2335
*s MediaChannels Call 4 OutgoingVideoChannel 45 Encryption Status: Off
*s MediaChannels Call 4 OutgoingVideoChannel 45 ChannelRole: Main
*s MediaChannels Call 4 OutgoingVideoChannel 45 Video Protocol: H264NIL
*s MediaChannels Call 4 OutgoingVideoChannel 45 Video FrameRate: 60
*s MediaChannels Call 4 OutgoingVideoChannel 45 Video ResolutionX: 1280
*s MediaChannels Call 4 OutgoingVideoChannel 45 Video ResolutionY: 720
*s MediaChannels Call 4 OutgoingVideoChannel 45 Transport RTP Local
IpAddress: "10.54.86.241"
*s MediaChannels Call 4 OutgoingVideoChannel 45 Transport RTP Local Port:
16404
*s MediaChannels Call 4 OutgoingVideoChannel 45 Transport RTP Remote
IpAddress: "10.54.86.240"
*s MediaChannels Call 4 OutgoingVideoChannel 45 Transport RTP Remote Port:
2336
*s MediaChannels Call 4 OutgoingVideoChannel 45 Transport RTCP Local
IpAddress: "10.54.86.241"
*s MediaChannels Call 4 OutgoingVideoChannel 45 Transport RTCP Local Port:
16405
*s MediaChannels Call 4 OutgoingVideoChannel 45 Transport RTCP Remote
IpAddress: "10.54.86.240"
*s MediaChannels Call 4 OutgoingVideoChannel 45 Transport RTCP Remote Port:
2337
*s MediaChannels Call 4 OutgoingVideoChannel 48 Encryption Status: Off
*s MediaChannels Call 4 OutgoingVideoChannel 48 ChannelRole: Presentation
*s MediaChannels Call 4 OutgoingVideoChannel 48 Video Protocol: Off
*s MediaChannels Call 4 OutgoingVideoChannel 48 Video FrameRate: 0
*s MediaChannels Call 4 OutgoingVideoChannel 48 Video ResolutionX: 0
*s MediaChannels Call 4 OutgoingVideoChannel 48 Video ResolutionY: 0
*s MediaChannels Call 4 OutgoingVideoChannel 48 Transport RTP Local
IpAddress: ""
*s MediaChannels Call 4 OutgoingVideoChannel 48 Transport RTP Local Port: 0
*s MediaChannels Call 4 OutgoingVideoChannel 48 Transport RTP Remote
IpAddress: ""
*s MediaChannels Call 4 OutgoingVideoChannel 48 Transport RTP Remote Port: 0
*s MediaChannels Call 4 OutgoingVideoChannel 48 Transport RTCP Local
IpAddress: ""
*s MediaChannels Call 4 OutgoingVideoChannel 48 Transport RTCP Local Port: 0
*s MediaChannels Call 4 OutgoingVideoChannel 48 Transport RTCP Remote
IpAddress: ""
*s MediaChannels Call 4 OutgoingVideoChannel 48 Transport RTCP Remote Port:
0
*s MediaChannels Call 4 OutgoingVideoChannel 52 Encryption Status: Off
*s MediaChannels Call 4 OutgoingVideoChannel 52 ChannelRole: Legacy
*s MediaChannels Call 4 OutgoingVideoChannel 52 Video Protocol: Off
*s MediaChannels Call 4 OutgoingVideoChannel 52 Video FrameRate: 0
*s MediaChannels Call 4 OutgoingVideoChannel 52 Video ResolutionX: 0
*s MediaChannels Call 4 OutgoingVideoChannel 52 Video ResolutionY: 0
*s MediaChannels Call 4 OutgoingVideoChannel 52 Transport RTP Local
IpAddress: ""
*s MediaChannels Call 4 OutgoingVideoChannel 52 Transport RTP Local Port: 0
*s MediaChannels Call 4 OutgoingVideoChannel 52 Transport RTP Remote
IpAddress: ""
*s MediaChannels Call 4 OutgoingVideoChannel 52 Transport RTP Remote Port: 0
*s MediaChannels Call 4 OutgoingVideoChannel 52 Transport RTCP Local
IpAddress: ""
*s MediaChannels Call 4 OutgoingVideoChannel 52 Transport RTCP Local Port: 0
*s MediaChannels Call 4 OutgoingVideoChannel 52 Transport RTCP Remote
IpAddress: ""
*s MediaChannels Call 4 OutgoingVideoChannel 52 Transport RTCP Remote Port:
0
** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Encryption Status

Shows the encryption status on the incoming channel.

Value space of the result returned:

<On/Off>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Encryption Status
*s MediaChannels Call 27 IncomingAudioChannel 327 Encryption Status: Off
** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Audio Protocol

Shows the audio algorithm for the incoming audio channel.

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

G722: The G.722 algorithm is an ITU standard.

G7221: The G.722.1 algorithm is a licensed royalty-free ITU-T standard.

G711Mu: The G.711 Mu-law compression algorithm is used in North America and Japan.

G711A: The G.711 A-law compression algorithm is used in Europe and the rest of the world

Value space of the result returned:

<AACLD/G722/G7221/G711Mu/G711A>

Example:

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

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Audio Mute

Audio mute status of incoming audio.

Value space of the result returned:

<True/False>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Audio Mute
*s MediaChannels Call 27 IncomingAudioChannel 327 Audio Mute: True
** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Audio Channels

Shows the number of incoming audio channels.

Value space of the result returned:

<Integer>

Example:

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

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Transport RTP Local IpAddress

Shows the local IP address of the Real-time Transport Protocol (RTP) port for the incoming audio in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Transport RTP Local
IpAddress
*s MediaChannels Call 27 IncomingAudioChannel 327 Transport RTP Local
IpAddress: "192.168.24.190"
** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Transport RTP Local Port

Shows the local UDP port number of the Real-time Transport Protocol (RTP) port for the incoming audio in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Transport RTP Local
Port
*s MediaChannels Call 27 IncomingAudioChannel 327 Transport RTP Local Port:
16404
** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Transport RTP Remote IpAddress

Shows the remote IP address of the Real-time Transport Protocol (RTP) port for the incoming audio in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Transport RTP Remote
IpAddress
*s MediaChannels Call 27 IncomingAudioChannel 327 Transport RTP Remote
IpAddress: "192.168.136.130"
** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Transport RTP Remote Port

Shows the remote UDP port number of the Real-time Transport Protocol (RTP) port for the incoming audio in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Transport RTP Remote
Port
*s MediaChannels Call 27 IncomingAudioChannel 327 Transport RTP Remote Port:
50932
** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Transport RTCP Local IpAddress

Shows the local IP address of the Real-time Transport Control Protocol (RTCP) port for the incoming audio in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Transport RTCP Local
IpAddress
*s MediaChannels Call 27 IncomingAudioChannel 327 Transport RTCP Local
IpAddress: "192.168.24.190"
** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Transport RTCP Local Port

Shows the local UDP port number of the Real-time Transport Control Protocol (RTCP) port for the incoming audio in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Transport RTCP Local
Port
*s MediaChannels Call 27 IncomingAudioChannel 327 Transport RTCP Local Port:
16405
** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Transport RTCP Remote IpAddress

Shows the remote IP address of the Real-time Transport Control Protocol (RTCP) port for the incoming audio in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Transport RTCP
Remote IpAddress
*s MediaChannels Call 27 IncomingAudioChannel 327 Transport RTCP Remote
IpAddress: "192.168.136.130"
** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Transport RCTP Remote Port

Shows the remote UDP port number of the Real-time Transport Control Protocol (RTCP) port for the incoming audio in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Transport RTCP
Remote Port
*s MediaChannels Call 27 IncomingAudioChannel 327 Transport RTCP Remote
Port: 50933
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Encryption Status

Shows the encryption status on the incoming channel.

Value space of the result returned:

<On/Off>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Encryption Status
*s MediaChannels Call 27 IncomingVideoChannel 330 Encryption Status: Off
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] ChannelRole

Shows if the incoming channel is the main video channel or presentation channel.

Value space of the result returned:

<Main/Presentation>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 ChannelRole
*s MediaChannels Call 27 IncomingVideoChannel 330 ChannelRole: Main
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Video Protocol

Shows the video algorithm for the incoming video channel.

H264: The H.264 algorithm is an ITU-T standard for video compression.

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

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

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

Value space of the result returned:

<H264/H263pp/H263/H261>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Video Protocol
*s MediaChannels Call 27 IncomingVideoChannel 330 Video Protocol: H264
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Video FrameRate

Shows the video frame rate of the incoming channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Video FrameRate
*s MediaChannels Call 27 IncomingVideoChannel 330 Video FrameRate: 25
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Video ResolutionX

Shows the width (resolution in direction X) of the incoming video.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Video ResolutionX
*s MediaChannels Call 27 IncomingVideoChannel 330 Video ResolutionX: 768
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Video ResolutionY

Shows the height (resolution in direction Y) of the incoming video.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Video ResolutionY
*s MediaChannels Call 27 IncomingVideoChannel 330 Video ResolutionY: 448
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Transport RTP Local IpAddress

Shows the local IP address of the Real-time Transport Protocol (RTP) port for the incoming video in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Transport RTP Local
IpAddress
*s MediaChannels Call 27 IncomingVideoChannel 330 Transport RTP Local
IpAddress: "192.168.24.190"
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Transport RTP Local Port

Shows the local UDP port number of the Real-time Transport Protocol (RTP) port for the incoming video in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Transport RTP Local
Port
*s MediaChannels Call 27 IncomingVideoChannel 330 Transport RTP Local Port:
16404
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Transport RTP Remote IpAddress

Shows the remote IP address of the Real-time Transport Protocol (RTP) port for the incoming video in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Transport RTP Remote
IpAddress
*s MediaChannels Call 27 IncomingVideoChannel 330 Transport RTP Remote
IpAddress: "192.168.136.130"
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Transport RTP Remote Port

Shows the remote UDP port number of the Real-time Transport Protocol (RTP) port for the incoming video in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Transport RTP Remote
Port
*s MediaChannels Call 27 IncomingVideoChannel 330 Transport RTP Remote Port:
50932
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Transport RTCP Local IpAddress

Shows the local IP address of the Real-time Transport Control Protocol (RTCP) port for the incoming video in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Transport RTCP Local
IpAddress
*s MediaChannels Call 27 IncomingVideoChannel 330 Transport RTCP Local
IpAddress: "192.168.24.190"
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Transport RTCP Local Port

Shows the local UDP port number of the Real-time Transport Control Protocol (RTCP) port for the incoming video in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Transport RTCP Local
Port
*s MediaChannels Call 27 IncomingVideoChannel 330 Transport RTCP Local Port:
16405
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Transport RTCP Remote IpAddress

Shows the remote IP address of the Real-time Transport Control Protocol (RTCP) port for the incoming video in the media channel..

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Transport RTCP
Remote IpAddress
*s MediaChannels Call 27 IncomingVideoChannel 330 Transport RTCP Remote
IpAddress: "192.168.136.130"
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Transport RCTP Remote Port

Shows the remote UDP port number of the Real-time Transport Control Protocol (RTCP) port for the incoming video in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Transport RTCP
Remote Port
*s MediaChannels Call 27 IncomingVideoChannel 330 Transport RTCP Remote
Port: 50933
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Encryption Status

Shows the encryption status on the outgoing channel.

Value space of the result returned:

<On/Off>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Encryption Status
*s MediaChannels Call 27 OutgoingAudioChannel 328 Encryption Status: Off
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Audio Protocol

Shows the audio algorithm for the outgoing audio channel.

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

G722: The G.722 algorithm is an ITU standard.

G7221: The G.722.1 algorithm is a licensed royalty-free ITU-T standard.

G711Mu: The G.711 Mu-law compression algorithm is used in North America and Japan.

G711A: The G.711 A-law compression algorithm is used in Europe and the rest of the world

Value space of the result returned:

<AACLD/G722/G7221/G711Mu/G711A>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Audio Protocol
*s MediaChannels Call 27 OutgoingAudioChannel 328 Audio Protocol: AACLD
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Audio Channels

Shows the number of outgoing audio channels.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Audio Channels
*s MediaChannels Call 27 OutgoingAudioChannel 328 Audio Channels: 1
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Transport RTP Local IpAddress

Shows the local IP address of the Real-time Transport Protocol (RTP) port for the outgoing audio in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTP Local
IpAddress
*s MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTP Local
IpAddress: "192.168.24.190"
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Transport RTP Local Port

Shows the local UDP port number of the Real-time Transport Protocol (RTP) port for the outgoing audio in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTP Local
Port
*s MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTP Local Port:
16404
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Transport RTP Remote IpAddress

Shows the remote IP address of the Real-time Transport Protocol (RTP) port for the outgoing audio in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTP Remote
IpAddress
*s MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTP Remote
IpAddress: "192.168.136.130"
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Transport RTP Remote Port

Shows the remote UDP port number of the Real-time Transport Protocol (RTP) port for the outgoing audio in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTP Remote
Port
*s MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTP Remote Port:
50932
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Transport RTCP Local IpAddress

Shows the local IP address of the Real-time Transport Control Protocol (RTCP) port for the outgoing audio in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTCP Local
IpAddress
*s MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTCP Local
IpAddress: "192.168.24.190"
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Transport RTCP Local Port

Shows the local UDP port number of the Real-time Transport Control Protocol (RTCP) port for the outgoing audio in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTCP Local
Port
*s MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTCP Local Port:
16405
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Transport RTCP Remote IpAddress

Shows the remote IP address of the Real-time Transport Control Protocol (RTCP) port for the outgoing audio in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTCP
Remote IpAddress
*s MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTCP Remote
IpAddress: "192.168.136.130"
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Transport RCTP Remote Port

Shows the remote UDP port number of the Real-time Transport Control Protocol (RTCP) port for the outgoing audio in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTCP
Remote Port
*s MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTCP Remote
Port: 50933
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Encryption Status

Shows the encryption status on the outgoing channel.

Value space of the result returned:

<On/Off>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Encryption Status
*s MediaChannels Call 27 OutgoingVideoChannel 331 Encryption Status: Off
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] ChannelRole

Shows if the outgoing channel is the main video channel or presentation channel.

Value space of the result returned:

<Main/Presentation>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 ChannelRole
*s MediaChannels Call 27 OutgoingVideoChannel 331 ChannelRole: Main
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Video Protocol

Shows the video algorithm for the outgoing video channel.

H264: The H.264 algorithm is an ITU-T standard for video compression.

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

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

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

Value space of the result returned:

<H264/H263pp/H263/H261>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video Protocol
*s MediaChannels Call 27 OutgoingVideoChannel 331 Video Protocol: "H264"
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Video FrameRate

Shows the video frame rate of the outgoing channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video FrameRate
*s MediaChannels Call 27 OutgoingVideoChannel 331 Video FrameRate: 30
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Video ResolutionX

Shows the width (resolution in direction X) of the outgoing video.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video ResolutionX
*s MediaChannels Call 27 OutgoingVideoChannel 331 Video ResolutionX: 768
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Video ResolutionY

Shows the height (resolution in direction Y) of the outgoing video.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video ResolutionY
*s MediaChannels Call 27 OutgoingVideoChannel 331 Video ResolutionY: 448
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Transport RTP Local IpAddress

Shows the local IP address of the Real-time Transport Protocol (RTP) port for the outgoing video in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTP Local
IpAddress
*s MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTP Local
IpAddress: "192.168.24.190"
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Transport RTP Local Port

Shows the local UDP port number of the Real-time Transport Protocol (RTP) port for the outgoing video in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTP Local
Port
*s MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTP Local Port:
16404
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Transport RTP Remote IpAddress

Shows the remote IP address of the Real-time Transport Protocol (RTP) port for the outgoing video in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTP Remote
IpAddress
*s MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTP Remote
IpAddress: "192.168.136.130"
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Transport RTP Remote Port

Shows the remote UDP port number of the Real-time Transport Protocol (RTP) port for the outgoing video in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTP Remote
Port
*s MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTP Remote Port:
50932
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Transport RTCP Local IpAddress

Shows the local IP address of the Real-time Transport Control Protocol (RTCP) port for the outgoing video in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTCP Local
IpAddress
*s MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTCP Local
IpAddress: "192.168.24.190"
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Transport RTCP Local Port

Shows the local UDP port number of the Real-time Transport Control Protocol (RTCP) port for the outgoing video in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTCP Local
Port
*s MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTCP Local Port:
16405
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Transport RTCP Remote IpAddress

Shows the remote IP address of the Real-time Transport Control Protocol (RTCP) port for the outgoing video in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTCP Remote
IpAddress
*s MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTCP Remote
IpAddress: "192.168.136.130"
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Transport RCTP Remote Port

Shows the remote UDP port number of the Real-time Transport Control Protocol (RTCP) port for the outgoing video in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTCP Remote
Port
*s MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTCP Remote
Port: 50933
** end
```

The Network status

xStatus Network

Shows the top level overview of the network status.

Example:

```
xStatus Network
*s Network 1 Ethernet MacAddress: "00:50:60:02:E7:D3"
*s Network 1 Ethernet Speed: "1000full"
*s Network 1 IPv4 Address: "192.0.2.149"
*s Network 1 IPv4 SubnetMask: "255.255.255.0"
*s Network 1 IPv4 Gateway: "192.0.2.10"
*s Network 1 IPv4 DNS Domain Name: "www.example.com www.example.int"
*s Network 1 IPv4 DNS Server 1 Address: "192.0.2.60"
*s Network 1 IPv4 DNS Server 2 Address: "192.0.2.61"
*s Network 1 IPv4 DNS Server 3 Address: ""
*s Network 1 IPv4 DNS Server 4 Address: ""
*s Network 1 IPv4 DNS Server 5 Address: ""
*s Network 1 IPv6 Address: ""
*s Network 1 IPv6 Gateway: ""
*s Network 1 IPv4 MTU: 1500
*s Network 1 VLAN Voice VlanId: "Off"
** 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 SubnetMask

Shows the subnet mask which determines which subnet an IPv4 address belongs to.

Value space of the result returned:

<String>

Example:

```
xStatus Network 1 IPv4 SubnetMask
*s Network 1 IPv4 SubnetMask: "255.255.255.0"
** end
```

xStatus Network 1 IPv4 Gateway

Shows the address of the IPv4 gateway.

Value space of the result returned:

<String>

Example:

```
xStatus Network 1 IPv4 Gateway
*s Network 1 IPv4 Gateway: "192.0.2.10"
** end
```

xStatus Network 1 IPv4 DNS Domain Name

Shows the domain name.

Value space of the result returned:

<String>

Example:

```
xStatus Network 1 IPv4 DNS Domain Name
*s Network 1 IPv4 DNS Domain Name: "www.example.com www.example.int"
** end
```

xStatus Network 1 IPv4 DNS Server [1..5] Address

Shows the IP address of the DNS server.

Value space of the result returned:

<String>

Example:

```
xStatus Network 1 IPv4 DNS Server 1. Address
*s Network 1 IPv4 DNS Server 1 Address: "192.0.2.60"
** end
```

xStatus Network 1 IPv6 Address

Shows the IPv6 address that uniquely identifies this system.

Value space of the result returned:

<String>

Example:

```
xStatus Network 1 IPv6 Address
*s Network 1 IPv6 Address: ""
** end
```

xStatus Network 1 IPv6 Gateway

Shows the address of the IPv6 gateway.

Value space of the result returned:

<String>

Example:

```
xStatus Network 1 IPv6 Gateway
*s Network 1 IPv6 Gateway: ""
** end
```

xStatus Network 1 MTU

Shows the MTU (Maximum Transmission Unit) size for the network.

Value space of the result returned:

<Integer>

Example:

```
xStatus Network 1 MTU
*s Network 1 MTU: 1500
** end
```

xStatus Network 1 VLAN Voice VlanId

The feedback will show 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
```

The Preset status

xStatus Preset

Shows the top level overview of the camera presets status.

Example:

```
xStatus Preset
*s Preset 1 Defined: True
*s Preset 1 Type: All
*s Preset 1 Description: "Zoom in"
*s Preset 2 Defined: True
*s Preset 2 Type: All
*s Preset 2 Description: "Zoom out"
 -
 - //continues with Preset 3-15.//
** end
```

xStatus Preset [1..15] Defined

Shows if a camera preset is stored at this position.

Value space of the result returned:

<True/False>

Example:

```
xStatus Preset 1 Defined
*s Preset 1 Defined: True
** end
```

xStatus Preset [1..15] Type

Shows the camera preset type.

Value space of the result returned:

<All/Camera>

Example:

```
xStatus Preset 1 Type
*s Preset 1 Type: All
** end
```

xStatus Preset [1..15] Description

Lists the configured name for the specific preset.

Value space of the result returned:

<String>

Example:

```
xStatus Preset 1 Description
*s Preset 1 Description: "Zoom in"
** end
```

The Provisioning status

xStatus Provisioning

Shows the top level overview of the provisioning status.

Example:

```
xStatus Provisioning
*s Provisioning Status: Provisioned
*s Provisioning Reason: ""
*s Provisioning Software UpgradeStatus SessionId: ""
*s Provisioning Software UpgradeStatus LastChange: "2011-06-07T07:20:03Z"
*s Provisioning Software UpgradeStatus Status: None
*s Provisioning Software UpgradeStatus Phase: None
*s Provisioning Software UpgradeStatus Message: ""
*s Provisioning Software UpgradeStatus VersionId: ""
*s Provisioning Software UpgradeStatus URL: ""
*s Provisioning Software Current VersionId: ""
*s Provisioning Software Current URL: ""
*s Provisioning Software Current CompletedAt: "2011-06-07T07:20:03Z"
** 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.

Value space of the result returned:

<Failed/AuthenticationFailed/Provisioned/Idle/NeedConfig>

Example:

```
xStatus Provisioning Status
*s Provisioning Status: Provisioned
** end
```

xStatus Provisioning Reason

Shows the cause when provisioning has failed.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning Reason
*s Provisioning Reason: ""
** end
```

xStatus Provisioning Software UpgradeStatus SessionId

Shows the ID of the session for the software upgrade.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning Software UpgradeStatus SessionId
*s Provisioning Software UpgradeStatus SessionId: ""
** end
```

xStatus Provisioning Software UpgradeStatus LastChange

Shows the date and time for the latest software upgrade.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning Software UpgradeStatus LastChange
*s Provisioning Software UpgradeStatus LastChange: "2011-06-07T07:20:03Z"
** end
```

xStatus Provisioning Software UpgradeStatus Status

Shows the status of the software upgrade.

Value space of the result returned:

<None/InProgress/Failed/Succeeded>

Example:

```
xStatus Provisioning Software UpgradeStatus Status
*s Provisioning Software UpgradeStatus Status: None
** end
```

xStatus Provisioning Software UpgradeStatus Phase

Shows the phase of the software upgrade.

Value space of the result returned:

<None/DownloadPending/Downloading/Installing>

Example:

```
xStatus Provisioning Software UpgradeStatus Phase
*s Provisioning Software UpgradeStatus Phase: None
** end
```

xStatus Provisioning Software UpgradeStatus Message

Shows the system message for the software upgrade.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning Software UpgradeStatus Message
*s Provisioning Software UpgradeStatus Message: ""
** end
```

xStatus Provisioning Software UpgradeStatus VersionId

Shows the version ID of the software currently being uploaded and installed.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning Software UpgradeStatus VersionId
*s Provisioning Software UpgradeStatus VersionId: "s52000tc5_1_0.pkg"
** end
```

xStatus Provisioning Software UpgradeStatus URL

Shows the URL that the new software currently is being uploaded and installed from.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning Software UpgradeStatus URL
*s Provisioning Software UpgradeStatus URL: "http://.../s52000tc5_1_0.pkg"
** end
```

xStatus Provisioning Software Current VersionId

Shows the version ID of the current software.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning Software Current VersionId
*s Provisioning Software Current VersionId: "s52000tc5 _ 1 _ 0.pkg"
** end
```

xStatus Provisioning Software Current URL

Shows the URL that the current software was uploaded from.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning Software Current URL
*s Provisioning Software Current URL: "http://.../s52000tc5 _ 1 _ 0.pkg"
** end
```

xStatus Provisioning Software Current CompletedAt

Shows date and time for when the current software upgrade was completed.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning Software Current CompletedAt
*s Provisioning Software Current CompletedAt: "2011-06-07T07:20:03Z"
** end
```

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
```

The SIP status

xStatus SIP

Shows the top level overview of the SIP status.

Example:

```
xStatus SIP
*s SIP Proxy 1 Status: Active
*s SIP Proxy 1 Address: "192.0.2.50"
*s SIP Proxy 1 Secure: True
*s SIP Proxy 1 Verified: False
*s SIP Registration 1 Status: Registered
*s SIP Registration 1 Reason: ""
*s SIP Registration 1 URI: "anyname@company.com"
*s SIP Registration 1 Authentication: Off
*s SIP Profile 1 Proxy 1 Status: Active
*s SIP Profile 1 Proxy 1 Address: "192.0.1.50"
*s SIP Profile 1 Secure: True
*s SIP Profile 1 Verified: False
*s SIP Profile 1 Authentication: Off
*s SIP Profile 1 Registration 1 Status: Registered
*s SIP Profile 1 Registration 1 Reason: ""
*s SIP Profile 1 Registration 1 URI: "anyname@company.com"
** end
```

xStatus SIP Proxy [1] Status

Shows the status of the communication between the endpoint and the SIP Proxy server.

Active: The communication between the endpoint and the SIP Proxy is active.

DNSFailed: The attempt to establish communication to the DNS server failed.

Off: There is no communication between the endpoint and the SIP Proxy.

Timeout: The attempt to establish communication to the SIP Proxy timed out.

UnableTCP: The system is unable to use TCP as the transport method.

UnableTLS: The system is unable to use TLS as the transport method.

Unknown: The status of the communication is not known.

Value space of the result returned:

<Active/DNSFailed/Off/Timeout/UnableTCP/UnableTLS/Unknown>

Example:

```
xStatus SIP Proxy 1 Status
*s SIP Proxy 1 Status: Active
** end
```

xStatus SIP Proxy [1] Address

Shows the address of the SIP Proxy that the system communicates with.

Value space of the result returned:

<String>

Example:

```
xStatus SIP Proxy 1 Address
*s SIP Proxy 1 Address: "192.0.2.50"
** end
```

xStatus SIP Proxy [1] Secure

Shows the encryption status of the signalling with the SIP Proxy server.

Value space of the result returned:

<True/False>

Example:

```
xStatus SIP Proxy 1 Secure
*s SIP Proxy 1 Secure: True
** end
```

xStatus SIP Proxy [1] Verified

Not supported in this software version.

Value space of the result returned:

<True/False>

Example:

```
xStatus SIP Proxy 1 Verified
*s SIP Proxy 1 Verified: False
** end
```

xStatus SIP Registration [1..n] Status

Shows the status of the registration to the SIP Proxy Server.

Deregister: The system is in the process of de-registering to the SIP Proxy.

Failed: The system failed to register to the SIP Proxy.

Inactive: The system is not registered to any SIP Proxy.

Registered: The system is registered to the SIP Proxy.

Registering: The system is in the process of registering to the SIP Proxy.

Value space of the result returned:

<Deregister/Failed/Inactive/Registered/Registering>

Example:

```
xStatus SIP Registration 1 Status
*s SIP Registration 1 Status: Registered
** end
```

xStatus SIP Registration [1..n] Reason

Shows a message to explain the reason why the SIP registration failed.

Value space of the result returned:

<String>

Example:

```
xStatus SIP Registration 1 Reason
*s SIP Registration 1 Reason: "404 Not Found"
** end
```

xStatus SIP Registration [1..n] URI

Shows the URI used for registration to the SIP Proxy server.

Value space of the result returned:

<String>

Example:

```
xStatus SIP Registration 1 URI
*s SIP Registration 1 URI: "firstname.lastname@company.com"
** end
```

xStatus SIP Registration [1..n] Authentication

Shows which authentication mechanism is used when registering to the SIP Proxy Server.

Digest: Uses the Digest access authentication method, as specified by RFC 2069.

NTLM: Uses the NTLM athentication method, which is a Microsoft athentication protocol.

Off: No authentication mechanism is used.

Value space of the result returned:

<Digest/NTLM/Off>

Example:

```
xStatus SIP Registration 1 Authentication
*s SIP Registration 1 Authentication: Off
** end
```

xStatus SIP Profile 1 Proxy [1] Status

Shows the status of the communication between the endpoint and the SIP Proxy server.

Active: The communication between the endpoint and the SIP Proxy is active.

DNSFailed: The attempt to establish communication to the DNS server failed.

Off: There is no communication between the endpoint and the SIP Proxy.

Timeout: The attempt to establish communication to the SIP Proxy timed out.

UnableTCP: The system is unable to use TCP as the transport method.

UnableTLS: The system is unable to use TLS as the transport method.

Unknown: The status of the communication is not known.

Value space of the result returned:

<Active/DNSFailed/Off/Timeout/UnableTCP/UnableTLS/Unknown>

Example:

```
xStatus SIP Profile 1 Proxy 1 Status
*s SIP Profile 1 Proxy 1 Status: Active
** end
```

xStatus SIP Profile 1 Proxy [1] Address

Shows the address of the SIP Proxy that the system communicates with.

Value space of the result returned:

<String>

Example:

```
xStatus SIP Profile 1 Proxy 1 Address
*s SIP Profile 1 Proxy 1 Address: "192.0.2.50"
** end
```

xStatus SIP Profile 1 Secure

Shows the encryption status of the signalling with the SIP Proxy server.

Value space of the result returned:

<True/False>

Example:

```
xStatus SIP Profile 1 Secure
*s SIP Profile 1 Secure: True
** end
```

xStatus SIP Profile 1 Verified

Not supported in this software version.

Value space of the result returned:

<True/False>

Example:

```
xStatus SIP Profile 1 Verified
*s SIP Profile 1 Verified: False
** end
```

xStatus SIP Profile 1 Authentication

Shows which authentication mechanism is used when registering to the SIP Proxy Server.

Digest: Uses the Digest access authentication method, as specified by RFC 2069.

NTLM: Uses the NTLM athentication method, which is a Microsoft athentication protocol.

Off: No authentication mechanism is used.

Value space of the result returned:

<Digest/NTLM/Off>

Example:

```
xStatus SIP Profile 1 Authentication
*s SIP Profile 1 Authentication: Off
** end
```

xStatus SIP Profile 1 Registration [1..n] Status

Shows the status of the registration to the SIP Proxy Server.

Deregister: The system is in the process of de-registering to the SIP Proxy.

Failed: The system failed to register to the SIP Proxy.

Inactive: The system is not registered to any SIP Proxy.

Registered: The system is registered to the SIP Proxy.

Registering: The system is in the process of registering to the SIP Proxy.

Value space of the result returned:

<Deregister/Failed/Inactive/Registered/Registering>

Example:

```
xStatus SIP Profile 1 Registration 1 Status
*s SIP Profile 1 Registration 1 Status: Registered
** end
```

xStatus SIP Profile 1 Registration [1..n] Reason

Shows a message to explain the reason why the SIP registration failed.

Value space of the result returned:

<String>

Example:

```
xStatus SIP Profile 1 Registration 1 Reason
*s SIP Profile 1 Registration 1 Reason: "404 Not Found"
** end
```

xStatus SIP Profile 1 Registration [1..n] URI

Shows the URI used for registration to the SIP Proxy server.

Value space of the result returned:

<String>

Example:

```
xStatus SIP Profile 1 Registration 1 URI
*s SIP Profile 1 Registration 1 URI: "firstname.lastname@company.com"
** end
```

The Standby status

xStatus Standby Active

Shows if the system is in standby or not.

Value space of the result returned:

<On/Off>

Example:

```
xStatus Standby Active
*s Standby Active: Off
** end
```

The SystemUnit status

xStatus SystemUnit

Shows the top level overview of the system unit status.

Example:

```
xStatus SystemUnit
*s SystemUnit ProductType: "Cisco TelePresence Codec"
*s SystemUnit ProductId: "Cisco TelePresence Codec C60"
*s SystemUnit ProductPlatform: "C60"
*s SystemUnit Uptime: 864143
*s SystemUnit Software Application: "Endpoint"
*s SystemUnit Software Version: "TC4.2.0"
*s SystemUnit Software Name: "s52000"
*s SystemUnit Software ReleaseDate: "2011-06-03"
*s SystemUnit Software MaxVideoCalls: 3
*s SystemUnit Software MaxAudioCalls: 4
*s SystemUnit Software ReleaseKey: "true"
*s SystemUnit Software OptionKeys NaturalPresenter: "true"
*s SystemUnit Software OptionKeys MultiSite: "true"
*s SystemUnit Software OptionKeys PremiumResolution: "true"
*s SystemUnit Software OptionKeys DualDisplay: "true"
*s SystemUnit Hardware Module SerialNumber: "F9AA99A00090"
*s SystemUnit Hardware Module Identifier: "0"
*s SystemUnit Hardware Module CompatibilityLevel: "0"
*s SystemUnit Hardware MainBoard SerialNumber: "PH0999998"
*s SystemUnit Hardware MainBoard Identifier: "101701-3 [04]"
*s SystemUnit Hardware VideoBoard SerialNumber: "PH0999999"
*s SystemUnit Hardware VideoBoard Identifier: "101560-1 [02]"
*s SystemUnit Hardware BootSoftware: "U-Boot 2010.06-81"
*s SystemUnit Hardware MonitoringSoftware: "39"
*s SystemUnit Hardware Monitoring Fan 1 Status: "locked on 1096 rpm"
*s SystemUnit Hardware Monitoring Fan 2 Status: "locked on 1096 rpm"
*s SystemUnit Hardware Monitoring Fan 3 Status: "locked on 1096 rpm"
*s SystemUnit Hardware Monitoring Fan 4 Status: "locked on 1096 rpm"
*s SystemUnit Hardware Temperature: "64.0"
*s SystemUnit State System: Initialized
*s SystemUnit State MaxNumberOfCalls: 3
*s SystemUnit State MaxNumberOfActiveCalls: 3
*s SystemUnit State NumberOfActiveCalls: 1
*s SystemUnit State NumberOfSuspendedCalls: 0
```

```
*s SystemUnit State NumberOfInProgressCalls: 0
*s SystemUnit State Subsystem Application: Initialized
*s SystemUnit ContactInfo: "anyname@company.com"
** end
```

xStatus SystemUnit ProductType

Shows the product type.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit ProductType
*s SystemUnit ProductType: "Cisco TelePresence Codec"
** end
```

xStatus SystemUnit ProductId

Shows the product identity.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit ProductId
*s SystemUnit ProductId: "Cisco TelePresence Codec C90"
** end
```

xStatus SystemUnit ProductPlatform

Shows the product platform.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit ProductPlatform
*s SystemUnit ProductPlatform: "C90"
** end
```

xStatus SystemUnit Uptime

Shows the number of seconds since the last restart of the codec.

Value space of the result returned:

<Integer>

Example:

```
xStatus SystemUnit Uptime
*s SystemUnit Uptime: 597095
** end
```

xStatus SystemUnit Software Application

Shows which software application is running on the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software Application
*s SystemUnit Software Application: "Endpoint"
** end
```

xStatus SystemUnit Software Version

Shows the software version installed on the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software Version
*s SystemUnit Software Version: "TC5.1.0"
** end
```

xStatus SystemUnit Software Name

Shows the name of the software that is installed on the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software Name
*s SystemUnit Software Name: "s52000"
** end
```

xStatus SystemUnit Software ReleaseDate

Shows the release date of the software installed on the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software ReleaseDate
*s SystemUnit Software ReleaseDate: "2012-02-22"
** end
```

xStatus SystemUnit Software MaxVideoCalls

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

Value space of the result returned:

<Integer>

Example:

```
xStatus SystemUnit Software MaxVideoCalls
*s SystemUnit Software MaxVideoCalls: 3
** end
```

xStatus SystemUnit Software MaxAudioCalls

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

Value space of the result returned:

<Integer>

Example:

```
xStatus SystemUnit Software MaxAudioCalls
*s SystemUnit Software MaxAudioCalls: 4
** end
```

xStatus SystemUnit Software ReleaseKey

Shows if there is a valid releasekey for the software version that is installed on the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software ReleaseKey
*s SystemUnit Software ReleaseKey: "true"
** end
```

xStatus SystemUnit Software OptionKeys NaturalPresenter

Shows if the system has the option key installed that supports the NaturalPresenter functionality.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software OptionKeys NaturalPresenter
*s SystemUnit Software OptionKeys NaturalPresenter: "true"
** end
```

xStatus SystemUnit Software OptionKeys MultiSite

Shows if the system has the option key installed that supports the MultiSite functionality.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software OptionKeys MultiSite
*s SystemUnit Software OptionKeys MultiSite: "true"
** end
```

xStatus SystemUnit Software OptionKeys PremiumResolution

Shows if the system has the option key installed that supports the PremiumResolution functionality.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software OptionKeys PremiumResolution
*s SystemUnit Software OptionKeys PremiumResolution: "true"
** end
```

xStatus SystemUnit Software OptionKeys DualDisplay

Shows if the system has the option key installed that supports the DualDisplay functionality (applicable for C40 only).

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software OptionKeys DualDisplay
*s SystemUnit Software OptionKeys DualDisplay: "true"
** end
```

xStatus SystemUnit Hardware Module SerialNumber

Shows the serial number of the hardware module in the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Hardware Module SerialNumber
*s SystemUnit Hardware Module SerialNumber: "F9AA99A00090"
** end
```

xStatus SystemUnit Hardware Module Identifier

Shows the revision of the hardware module in the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Hardware Module Identifier
*s SystemUnit Hardware Module Identifier: "1"
** end
```

xStatus SystemUnit Hardware Module CompatibilityLevel

The Cisco TelePresence endpoints are using a NAND flash memory for general storage. The endpoints running TC software will be 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 will be 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 will not have the new flash memory installed.

The result returned when running the command will be either 0 or 1:

0 = The system does not have the new flash memory installed.

1 = The system has the new flash memory installed. If downgraded, it can only be downgraded to previous TC software versions having support for the new version of the flash memory.

Value space of the result returned:

<"1"/"0">

Example:

```
xstatus SystemUnit Hardware Module CompatibilityLevel
*s SystemUnit Hardware Module CompatibilityLevel: 1
** end
```

xStatus SystemUnit Hardware MainBoard SerialNumber

Shows the serial number of the main board in the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Hardware MainBoard SerialNumber
*s SystemUnit Hardware MainBoard SerialNumber: "PH0999989"
** end
```

xStatus SystemUnit Hardware MainBoard Identifier

Shows the revision of the main board in the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Hardware MainBoard Identifier
*s SystemUnit Hardware MainBoard Identifier: "101400-5 [06]"
** end
```

xStatus SystemUnit Hardware VideoBoard SerialNumber

Shows the serial number of the video board in the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Hardware VideoBoard SerialNumber
*s SystemUnit Hardware VideoBoard SerialNumber: "PH0999999"
** end
```

xStatus SystemUnit Hardware VideoBoard Identifier

Shows the revision of the video board in the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Hardware VideoBoard Identifier
*s SystemUnit Hardware VideoBoard Identifier: "101410-4 [07]"
** end
```

xStatus SystemUnit Hardware BootSoftware

Shows the version of the boot software that is installed on the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Hardware BootSoftware
*s SystemUnit Hardware BootSoftware: "U-Boot 2010.04-30"
** end
```

xStatus SystemUnit Hardware MonitoringSoftware

The feedback will show the monitoring software id.

Value space of the result returned:

<String>

Example:

```
xstatus SystemUnit Hardware MonitoringSoftware
*s SystemUnit Hardware MonitoringSoftware: "39"
** end
```

xStatus SystemUnit Hardware Monitoring Fan [1..n] Status

The feedback will show the speed (rpm) for the specified fan.

Value space of the result returned:

<String>

Example:

```
xstatus SystemUnit Hardware Monitoring Fan 1 Status
*s SystemUnit Hardware Monitoring Fan 1 Status: "locked on 1096 rpm"
** end
```

xStatus SystemUnit Hardware Temperature

The feedback will show 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 State System

Shows what state the system is in.

InCall: The system is in a call.

Initialized: The system is ready for use.

Initializing: The system is initializing.

Multisite: The system is in a Multisite conference.

Sleeping: The system is in sleep mode.

Value space of the result returned:

<InCall/Initialized/Initializing/Multisite/Sleeping>

Example:

```
xStatus SystemUnit State System
*s SystemUnit State System: Initialized
** end
```

xStatus SystemUnit State MaxNumberOfCalls

Shows the the maximum number of simultaneous calls.

Value space of the result returned:

<0..5>

Example:

```
xStatus SystemUnit State MaxNumberOfCalls
*s SystemUnit State MaxNumberOfCalls: 3
** end
```

xStatus SystemUnit State MaxNumberOfActiveCalls

Shows the the maximum number of simultaneous active calls. Calls that are set on hold/transfer are not counted as active.

Value space of the result returned:

<0..5>

Example:

```
xStatus SystemUnit State MaxNumberOfActiveCalls
*s SystemUnit State MaxNumberOfActiveCalls: 3
** end
```

xStatus SystemUnit State NumberOfActiveCalls

Shows the number of active calls.

Value space of the result returned:

<0..5>

Example:

```
xStatus SystemUnit State NumberOfActiveCalls
*s SystemUnit State NumberOfActiveCalls: 0
** end
```

xStatus SystemUnit State NumberOfSuspendedCalls

Shows the number of suspended calls.

Value space of the result returned:

<0..5>

Example:

```
xStatus SystemUnit State NumberOfSuspendedCalls
*s SystemUnit State NumberOfSuspendedCalls: 0
** end
```

xStatus SystemUnit State NumberOfInProgressCalls

Shows the number of calls in progress.

Value space of the result returned:

<0..5>

Example:

```
xStatus SystemUnit State NumberOfInProgressCalls
*s SystemUnit State NumberOfInProgressCalls: 0
** end
```

xStatus SystemUnit State Subsystem Application

Shows the status of the sub system application.

Initialized: The sub-system application is initialized.

Initializing: The sub-system application is initializing.

Value space of the result returned:

<Initialized/Initializing>

Example:

```
xStatus SystemUnit State Subsystem Application
*s SystemUnit State Subsystem Application: Initialized
** end
```

xStatus SystemUnit ContactInfo

Shows the address which another system can dial to reach this system.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit ContactInfo
*s SystemUnit ContactInfo: "firstname.lastname@company.com"
** end
```

xStatus SystemUnit Notifications Notification [1..n] Type

Lists the system notification types. Notifications are issued e.g. when a system was rebooted because of a software upgrade, or when a factory reset has been performed.

FactoryResetOK: This value is returned after a successful factory reset.

FactoryResetFailed: This value is returned after a failed factory reset attempt.

SoftwareUpgradeOK: This value is returned after a successful software upgrade.

SoftwareUpgradeFailed: This value is returned after a failed software upgrade attempt.

RebootRequired: This value is returned when a reboot is required.

Other: This value is returned for any other notifications.

All the notifications can be removed from the list by issuing the xCommand SystemUnit Notifications RemoveAll command.

Value space of the result returned:

<FactoryResetOK, FactoryResetFailed, SoftwareUpgradeOK, SoftwareUpgradeFailed, RebootRequired, Other>

Example:

```
xStatus SystemUnit Notifications Notification 1 Type
*s SystemUnit Notifications Notification 1 Type: SoftwareUpgradeOK
** end
```

xStatus SystemUnit Notifications Notification [1..n] Text

Lists text related to important system notifications. Notifications are issued e.g. when a system was rebooted because of a software upgrade, or when a factory reset has been performed.

All the notifications can be removed from the list by issuing the xCommand SystemUnit Notifications RemoveAll command.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Notifications Notification 1 Text
*s SystemUnit Notifications Notification 1 Text: "OK"
** end
```

The Time status

xStatus Time ZoneOlson

Shows the current time zone on Olson format.

Value space of the result returned:

<String>

Example:

```
xStatus Time ZoneOlson
*s Time ZoneOlson: Europe/Berlin
** end
```

The Video status

xStatus Video Input

Shows the top level overview of the video input status.

Example:

```
xStatus Video Input
*s Video Input LastConnectedSource: 0
*s Video Input Source 1 Resolution Height: 1080
*s Video Input Source 1 Resolution Width: 1920
*s Video Input Source 1 Resolution RefreshRate: 64
*s Video Input Source 1 Resolution FormatType: Digital
*s Video Input Source 1 Resolution FormatStatus: Ok
*s Video Input Source 2 Resolution Height: 0
*s Video Input Source 2 Resolution Width: 0
*s Video Input Source 2 Resolution RefreshRate: 0
*s Video Input Source 2 Resolution FormatType: Unknown
*s Video Input Source 2 Resolution FormatStatus: Error
*s Video Input Source 3 Resolution Height: 0
*s Video Input Source 3 Resolution Width: 0
*s Video Input Source 3 Resolution RefreshRate: 0
*s Video Input Source 3 Resolution FormatType: Unknown
*s Video Input Source 3 Resolution FormatStatus: Error
*s Video Input HDMI 1 Connected: True
*s Video Input HDMI 1 SignalState: OK
*s Video Input HDMI 2 Connected: True
*s Video Input HDMI 2 SignalState: OK
*s Video Input DVI 2 Connected: Unknown
*s Video Input DVI 2 SignalState: Unknown
*s Video Input DVI 3 Connected: Unknown
*s Video Input DVI 3 SignalState: Unknown
*s Video Input Legacy 3 Connected: False
*s Video Input Legacy 3 SignalState: Unknown
** end
```

xStatus Video Input LastConnectedSource

Shows the last connected video input source.

Value space of the result returned:

<1..3>

Example:

```
xStatus Video Input LastConnectedSource
*s Video Input LastConnectedSource: 0
** end
```

xStatus Video Input Source [1..3] Resolution Height

Shows the resolution height (in pixels) for the video input source.

Value space of the result returned:

<0..3000>

Example:

```
xStatus Video Input Source 1 Resolution Height
*s Video Input Source 1 Resolution Height: 1080
** end
```

xStatus Video Input Source [1..3] Resolution Width

Shows the resolution width (in pixels) for the video input source.

Value space of the result returned:

<0..4000>

Example:

```
xStatus Video Input Source 1 Resolution Width
*s Video Input Source 1 Resolution Width: 1920
** end
```

xStatus Video Input Source [1..3] Resolution RefreshRate

Shows the resolution refresh rate (Hz) for the video input source.

Value space of the result returned:

<0..300>

Example:

```
xStatus Video Input Source 1 Resolution RefreshRate
*s Video Input Source 1 Resolution RefreshRate: 50
** end
```

xStatus Video Input Source [1..3] Resolution FormatType

Shows the resolution format type for the video input source.

Value space of the result returned:

<Unknown/AnalogCVTBlanking/AnalogCVTReducedBlanking/AnalogGTFDefault/AnalogGTFSecondary/AnalogDiscreteTiming/AnalogDMTBlanking/AnalogCEABlanking/Digital>

Example:

```
xStatus Video Input Source 1 Resolution FormatType
*s Video Input Source 1 Resolution FormatType: Digital
** end
```

xStatus Video Input Source [1..3] Resolution FormatStatus

Shows the resolution format status for the video input source.

Value space of the result returned:

<Ok/AnalogOutOfRange/AnalogNotFound/Interlaced/Error/Unknown>

Example:

```
xStatus Video Input Source 1 Resolution FormatStatus
*s Video Input Source 1 Resolution FormatStatus: Ok
** end
```

xStatus Video Input HDMI [1..2] Connected

Shows if there is something connected to the HDMI connector. Not all connections can be detected.

Value space of the result returned:

<True/False/Unknown>

Example:

```
xStatus Video Input HDMI 1 Connected
*s Video Input HDMI 1 Connected: True
** end
```

xStatus Video Input HDMI [1..2] SignalState

Shows the signal state for the HDMI input.

Unknown: The signal format is unknown.

OK: A signal is detected and the signal format is supported.

Unsupported: A signal is detected, but the signal format is not supported.

Value space of the result returned:

<Unknown/OK/Unsupported>

Example:

```
xStatus Video Input HDMI 1 SignalState
*s Video Input HDMI 1 SignalState: OK
** end
```

xStatus Video Input DVI [2..3] Connected

Shows if there is something connected to the DVI connector. Not all connections can be detected.

Value space of the result returned:

<True/False/Unknown>

Example:

```
xStatus Video Input DVI 2 Connected
*s Video Input DVI 2 Connected: False
** end
```

xStatus Video Input DVI [2, 3] SignalState

Shows the signal state for the DVI-I input.

Unknown: The signal format is unknown.

OK: A signal is detected and the signal format is supported.

Unsupported: A signal is detected, but the signal format is not supported.

Value space of the result returned:

<Unknown/OK/Unsupported>

Example:

```
xStatus Video Input DVI 2 SignalState
*s Video Input DVI 2 SignalState: OK
** end
```

xStatus Video Input Legacy [3] Connected

Shows if there is something connected to the Legacy, which are the Y/Comp and C connectors. Not all connections can be detected.

Value space of the result returned:

<True/False/Unknown>

Example:

```
xStatus Video Input Legacy 3 Connected
*s Video Input Legacy 3 Connected: False
** end
```

xStatus Video Input Legacy [3] SignalState

Shows the signal state for the Legacy, which are the Y/Comp and C inputs.

Unknown: The signal format is unknown.

OK: A signal is detected and the signal format is supported.

Unsupported: A signal is detected, but the signal format is not supported.

Value space of the result returned:

<Unknown/OK/Unsupported>

Example:

```
xStatus Video Input Legacy 3 SignalState
*s Video Input Legacy 3 SignalState: OK
** end
```

xStatus Video Output

Shows the top level overview of the video output status.

Example:

```
xStatus Video Output
*s Video Output HDMI 1 Resolution Height: 720
*s Video Output HDMI 1 Resolution Width: 1280
*s Video Output HDMI 1 Resolution RefreshRate: 60
*s Video Output DVI 2 Resolution Height: 768
*s Video Output DVI 2 Resolution Width: 1024
*s Video Output DVI 2 Resolution RefreshRate: 60
*s Video Output Legacy 3 Resolution Height: 480
*s Video Output Legacy 3 Resolution Width: 720
*s Video Output Legacy 3 Resolution RefreshRate: 60
** end
```

xStatus Video Output HDMI [1] Resolution Height

Shows the resolution height (in pixels) for the video output HDMI.

Value space of the result returned:

<120..3000>

Example:

```
xStatus Video Output HDMI 1 Resolution Height
*s Video Output HDMI 1 Resolution Height: 720
** end
```

xStatus Video Output HDMI [1] Resolution Width

Shows the resolution width (in pixels) for the video output HDMI.

Value space of the result returned:

<176..4000>

Example:

```
xStatus Video Output HDMI 1 Resolution Width
*s Video Output HDMI 1 Resolution Width: 1280
** end
```

xStatus Video Output HDMI [1] Resolution RefreshRate

Shows the resolution refresh rate (Hz) for the video output HDMI.

Value space of the result returned:

<1..300>

Example:

```
xStatus Video Output HDMI 1 Resolution RefreshRate
*s Video Output HDMI 1 Resolution RefreshRate: 60
** end
```

xStatus Video Output DVI [2] Resolution Height

Shows the resolution height (in pixels) for the video output DVI.

Value space of the result returned:

<120..3000>

Example:

```
xStatus Video Output DVI 2 Resolution Height
*s Video Output DVI 2 Resolution Height: 720
** end
```

xStatus Video Output DVI [2] Resolution Width

Shows the resolution width (in pixels) for the video output DVI.

Value space of the result returned:

<176..4000>

Example:

```
xStatus Video Output DVI 2 Resolution Width
*s Video Output DVI 2 Resolution Width: 1280
** end
```

xStatus Video Output DVI [2] Resolution RefreshRate

Shows the resolution refresh rate (Hz) for the video output DVI.

Value space of the result returned:

<1..300>

Example:

```
xStatus Video Output DVI 2 Resolution RefreshRate
*s Video Output DVI 2 Resolution RefreshRate: 60
** end
```

xStatus Video Output Legacy [3] Resolution Height

Shows the resolution height (in pixels) for the video output Legacy (Composite).

Value space of the result returned:

<120..3000>

Example:

```
xStatus Video Output Legacy 3 Resolution Height
*s Video Output Legacy 3 Resolution Height: 480
** end
```

xStatus Video Output Legacy [3] Resolution Width

Shows the resolution width (in pixels) for the video output Legacy (Composite).

Value space of the result returned:

<176..4000>

Example:

```
xStatus Video Output Legacy 3 Resolution Width
*s Video Output Legacy 3 Resolution Width: 720
** end
```

xStatus Video Output Legacy [3] Resolution RefreshRate

Shows the resolution refresh rate (Hz) for the video output Legacy (Composite).

Value space of the result returned:

<1..300>

Example:

```
xStatus Video Output Legacy 3 Resolution RefreshRate
*s Video Output Legacy 3 Resolution RefreshRate: 60
** end
```

xStatus Video Layout

Shows the top level overview of the video layout status.

Example:

```
xStatus Video Layout
*s Video Layout PresentationView: "full"
*s Video Layout Site 1 Output 1 FamilyName: "speaker"
*s Video Layout Site 1 Output 1 FullFamilyName: "speaker-sv-on"
*s Video Layout Site 1 Output 1 GraphicName: "1top-1small"
*s Video Layout Site 1 Output 1 Frame 1 PositionX: 1333
*s Video Layout Site 1 Output 1 Frame 1 PositionY: 59
*s Video Layout Site 1 Output 1 Frame 1 Width: 7334
*s Video Layout Site 1 Output 1 Frame 1 Height: 7334
*s Video Layout Site 1 Output 1 Frame 1 Layer: 1
*s Video Layout Site 1 Output 1 Frame 1 VideoSourceType: "site"
*s Video Layout Site 1 Output 1 Frame 1 VideoSourceId: 27
*s Video Layout Site 1 Output 1 Frame 1 InputNumber: 1
*s Video Layout Site 1 Output 1 Frame 1 Filename: ""
 -
 - continues with the video layout status for the Sites [1..n], Outputs [1..n]
and Frames [1..n].
** end
```

xStatus Video Layout PresentationView

Returns information about the presentation view mode.

Value space of the result returned:

<Default/Maximized/Minimized>

Example:

```
xStatus Video Layout PresentationView
*s Video Layout PresentationView: "Default"
** end
```

xStatus Video Layout Site [1..n] Output [1..3] FamilyName

Shows the name of the video layout family.

Value space of the result returned:

<String>

Example:

```
xStatus Video Layout Site 1 Output 1 FamilyName
*s Video Layout Site 1 Output 1 FamilyName: "fullscreen"
** end
```

xStatus Video Layout Site [1..n] Output [1..3] FullFamilyName

Shows the name, included information about selfview on/off, for the video layout family.

Value space of the result returned:

<String>

Example:

```
xStatus Video Layout Site 1 Output 1 FullFamilyName
*s Video Layout Site 1 Output 1 FullFamilyName: "fullscreen-local-single-
camctrl"
** end
```

xStatus Video Layout Site [1..n] Output [1..3] GraphicName

Shows the name of the graphic layout. The name identifies the layout used right now at the specified output.

NOTE: Note that while the FamilyName is constant as the long as the configurations on the system does not change, the GraphicName varies depending on system state (the number of participants for instance).

Value space of the result returned:

<String>

Example:

```
xStatus Video Layout Site 1 Output 1 GraphicName
*s Video Layout Site 1 Output 1 GraphicName: "full-pip"
** end
```

xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] PositionX

Shows the horizontal position of the upper left corner of the frame.

Value space of the result returned:

<0..10000>

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 PositionX
*s Video Layout Site 1 Output 1 Frame 1 PositionX: 0
** end
```

xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] PositionY

Shows the vertical position of the upper left corner of the frame.

Value space of the result returned:

<0..10000>

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 PositionY
*s Video Layout Site 1 Output 1 Frame 1 PositionY: 0
** end
```

xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] Width

Shows the width of the frame.

Value space of the result returned:

<0..10000>

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 Width
*s Video Layout Site 1 Output 1 Frame 1 Width: 10000
** end
```

xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] Height

Shows the height of the frame.

Value space of the result returned:

<0..10000>

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 Height
*s Video Layout Site 1 Output 1 Frame 1 Height: 10000
** end
```

xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] Layer

Shows the layer of the frame.

Value space of the result returned:

<1..6>

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 Layer
*s Video Layout Site 1 Output 1 Frame 1 Layer: 1
** end
```

xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] VideoSourceType

Describes the video source type in the frame.

Value space of the result returned:

<String>

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 VideoSourceType
*s Video Layout Site 1 Output 1 Frame 1 VideoSourceType: "graphic"
** end
```

xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] VideoSourceId

Shows the video source Id which is used when adding or updating frames. See the xCommand Video Layout Frame Add and xCommand Video Layout Frame Update.

Value space of the result returned:

<1..3>

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 VideoSourceId
*s Video Layout Site 1 Output 1 Frame 1 VideoSourceId: 0
** end
```

xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] InputNumber

Shows the layout input number.

Value space of the result returned:

<1..3>

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 InputNumber
*s Video Layout Site 1 Output 1 Frame 1 InputNumber: 0
** end
```

xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] Filename

Shows the filename of the layout frame.

Value space of the result returned:

<String>

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 Filename
*s Video Layout Site 1 Output 1 Frame 1 Filename: "/user/posters/wallpaper.
png"
** end
```

The Experimental status

The Experimental status is for testing only and should not be used unless agreed with Cisco.
These settings are not documented and WILL change in later releases.

Chapter 6

Appendices

Startup script

You can add a startup script on the codec to execute certain commands from the API during boot up.

Adding a startup script

The startup script can be used to execute certain commands from the API during boot up. To enable this feature one must log in to the codec as root and follow the below points.

Login to the codec

1. Connect to the codec through the network, using a command line interface (ssh, telnet or scp) and login as `root`
2. Make a user directory using the following command: "`mkdir /user/scripts.d`"
3. Put an executable file (permission must be changed to executable) in this directory.

Example of the text in such a file:

```
#!/usr/bin/env tsh
xCommand Audio LocalInput Update InputId: 1 MixerMode:Fixed
```

The startup script file

- The file must start with the following sequence:

  `#!/usr/bin/env tsh`
- The file can contain any xCommand or xConfiguration command
- The system will execute the commands/configurations in sequence.
- The file can have any name as long as it is placed in this directory.
- For multiple commands you must use Unix end of line (LF). Windows end of line will not work.

Cisco TelePresence Remote Control

Function keys: Represents shortcuts and advanced functions. Each key reflects a soft key on screen.

Microphone: Press the key to toggle the microphones on/off.

Volume: Press the + or – on the key to adjust the codec volume.

Mute: Press the – on the key to mute the volume during an incoming call.

OK/Select: Press the ✓ key to confirm your choice or selection.

Phone book: Press the key to display the local Phone book.

Home: Press the key to key to show the menu on screen.

Call: Using the key:

INITIATE A CALL: Select a name from the Phone book or enter the name, number or URI and press the Call key to initiate the call.

SHORTCUT TO RECENT CALLS: Use the Call key as a shortcut to Recent Calls when the Call menu is not visible.

Clear: Press the C key to remove characters in a text field.

Waking up the system

Grab the remote control and make sure your hand touches the rubber line sensors going along both sides of the remote control.

or: Touch any key on the remote control.

Presentation: Press the key to show/hide a presentation.

Zoom: Press the + or – on the key to zoom the camera in and out.

Arrows:

- Up/Down: Use the ▲ and ▼ arrow keys to navigate in the menu.
- Arrow Right: Press the ▶ arrow key to expand the selected menu item or to move to the right in a text field.
- Arrow Left: Press the ◀ arrow key to go one step back in the menu or to move to the left in a text field.

Layout: Press the key to display the Layout menu, then select a view in the menu.

End call/Standby: Press the key to end a call, or when idle, press and hold the key to go into standby mode.

Alphanumeric keypad: Use the keypad in the same way as you would use a cellular phone.

- 0-9, a-z, period (.), @, space, *: Press a key repeatedly to toggle between the options displayed on each key.
- abc/123 #: Press the # key to toggle between touch tones mode (long press), lower case characters and numbers.

IR transmitter range (DIP switch setting)

The IR transmitter has a short and long range. Open the battery cover and remove the batteries to set the DIP switch.

- Short range (1 m): Move the DIP switch down.
- Longer range: Move the DIP switch up.

Remote control key map

The Cisco TelePresence Remote Control 5 has the following button codes and IR signal parameters.

You will find a one page overview of the remote control on the next page.

Button codes - Remote control 5

Dec	*Hex*	*Address*	*Button name*
0	0		
1	1	0	Number 1
2	2	0	Number 2
3	3	0	Number 3
4	4	0	Number 4
5	5	0	Number 5
6	6	0	Number 6
7	7	0	Number 7
8	8	0	Number 8
9	9	0	Number 9
10	0A	0	Number 0
11	0B	0	*
12	0C	0	#
13	0D		
14	0E		
15	0F		
16	10		
17	11	0	Presenter
18	12		
19	13		
20	14		
21	15		
22	16	0	Zoom out
23	17	0	Zoom in
24	18		
25	19	0	Volume down
26	1A	0	Volume up
27	1B	0	Microphone off
28	1C		
29	1D	0	Arrow up
30	1E	0	Arrow down
31	1F	0	Arrow left

Button codes - Remote control 5

Dec	*Hex*	*Address*	*Button name*
32	20	0	Arrow right
33	21	0	Ok
34	22	0	Call
35	23	0	End call
36	24	0	Phone book
37	25	0	Layout
38	26	0	Clear
39	27		
–	–		
41	29		
42	2A	0	Soft key 1
43	2B	0	Soft key 2
44	2C	0	Soft key 3
45	2D	0	Soft key 4
46	2E	0	Soft key 5
–	–		
–	–		
–	–		
–	–		
51	33	0	Home

IR Signal parameters

Name	*Description*
Protocol	Siemens SDA2208
Reference frequency	485 kHz
Address	4 and 7
IR wavelength	940 nm
IR carrier frequency	30 kHz

About Disconnect Cause Types

The following parameters are logged when a calll is disconnected. The disconnect cause types are used in disconnect events (xEvent) and also logged in xHistory CallLogs.

CauseValue	Proprietary. We recomend the use of CauseType and CauseCode.
CauseType	Describes why the call was disconnected. The value space is { OtherLocal, LocalDisconnect, UnknownRemoteSite, LocalBusy, LocalReject, InsufficientSecurity, OtherRemote, RemoteDisconnect, RemoteBusy, RemoteRejected, RemoteNoAnswer, CallForwarded, NetworkRejected }
CauseString	Describes the Cause Code.
CauseCode	The disconnect Cause Codes are defined in SIP and Q.850.
CauseOrigin	SIP, Q.850, internal

Examples:

```
xHistory CallLogs Call 694
...
*h xHistory CallLogs Call 694 DisconnectCauseValue: 2
*h xHistory CallLogs Call 694 DisconnectCause: "Normal"
*h xHistory CallLogs Call 694 DisconnectCauseType: RemoteDisconnect
*h xHistory CallLogs Call 694 DisconnectCauseCode: 16
*h xHistory CallLogs Call 694 DisconnectCauseOrigin: Q850
...
** end
```

xEvent DisconnectEvent

```
*e CallDisconnect CauseValue: 1
   CauseType: "LocalDisconnect"
   CauseString: ""
   OrigCallDirection: "outgoing"
   RemoteURI: "firstname.lastname@company.com"
   CallId: 89
   CauseCode: 0
   CauseOrigin: SIP
** end
```

On our web site you will find an overview of the worldwide Cisco contacts.
Go to: http://www.cisco.com/web/siteassets/contacts

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA 95134 USA