Application Programmer Interface (API) 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](http://www.cisco.com/go/telepresence/docs)

## Table of Contents

### Introduction
- About this guide ........................................... 5
- User documentation ...................................... 5
- What’s new in this version ............................... 6
- Software release notes .................................... 6
- Software download ........................................ 6
- User documentation ...................................... 6
- New features and improvements ........................ 6
- Cisco TelePresence Touch for C Series ............... 6
- New API commands ....................................... 7
- API commands that have changed ..................... 7
- API commands that are removed ...................... 7
- Experimental settings ................................... 7
- Other API commands .................................... 7

### About the API
- About the API ............................................. 9
- Basic Principles .......................................... 9
- The API-Engine .......................................... 9
- Structuring of Information .............................. 9
- Addressing Using XPath or SimplePath .............. 10
- Feedback .................................................. 10
- Connecting to the codec ............................... 11
- Accessing XACLI .......................................... 11
- Telnet/SSH login ........................................ 11
- Serial port login ........................................ 11
- Serial port configurations ............................. 11
- Reboot ..................................................... 11
- Hardware & Cabling (RS-232) ......................... 11
- Troubleshooting (RS-232) ............................. 11
- Value types and formats ................................ 12
- User commands .......................................... 13
- Main type of commands ................................ 14
  - Configuration type commands ........................ 14
  - Command type commands ............................ 14

### Status type commands ............................ 14
- Special commands ...................................... 14
- Feedback type command ............................... 14
- Preferences type command ............................. 14
- About xConfiguration ................................. 15
- xConfiguration operations ........................... 16
- Return result parameters .............................. 16
- xConfiguration Help .................................. 16
- xConfiguration Read ................................... 16
- xConfiguration Write .................................. 16
- About xCommand ....................................... 17
- xCommand operations ................................. 18
- Return result parameters .............................. 18
- xCommand Help ........................................ 18
- xCommand Write ...................................... 18
- About xHistory command .............................. 19
- Query status information ............................. 20
- About xEvent command ............................... 21
- About xFeedback ....................................... 22
- About xFeedback ....................................... 23
- About xPreferences ................................... 24
- The xPreferences output modes ..................... 24
- The SystemTools commands .......................... 25
- XML API service ....................................... 27

### Description of the xConfiguration commands
- Description of the xConfiguration commands ........ 29

### Description of the xCommand commands
- xCommands with parameters .......................... 68
## Description of the xStatus commands

- xStatus commands .................................................. 107

## Appendices

- Startup script ......................................................... 152
  - Adding a startup script ...................................... 152
- Cisco TelePresence Remote Control ....................... 153
- Remote control key map .......................................... 154
- Overview of the xConfiguration commands ............... 155
- Overview of the xCommand commands ..................... 159
- Overview of the xStatus commands ......................... 162
Chapter 1

Introduction
## About this guide

The purpose of this document is to introduce you to the Application Programmer Interface (API) in general and to 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
  - When using the Touch controller, ref. TC4.1 version of the user guide
  - When using the Remote Control, ref. TC4.0 version of the user guide
- 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 products using TC software

Download the user documentation

Go to: [http://www.cisco.com/go/telepresence/docs](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 API commands and new features in the TC4.1 software version.

Software release notes

For a complete overview of the news and changes, we recommend reading the Software Release Notes (TC4).


Software download

For software download go to: http://www.cisco.com/cisco/software/navigator.html?a=a&r=rpm

User documentation

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.

New features and improvements

Cisco TelePresence Touch for C Series

The Cisco TelePresence Touch is a touch based user interface that supports Cisco’s vision for a natural user experience. You can make video calls, share content, and access some advanced feature - all with a simple tap of the finger.

The Cisco TelePresence Touch for C Series is available for:
  - Cisco TelePresence System Codec C Series (C40, C60, C90)
  - Cisco TelePresence System Profile Series using Codec C Series
  - Cisco TelePresence System Quick Set C20

NOTE: Cisco TelePresence Touch for C Series requires software version TC4.1.0 or later.

Supported features:
- Calling
- Call control
- Conferencing
- Presenting
- Presentation source selection
- Layout handling
- Directory handling
- Favorites list
- Call history management
- Basic system configuration
- Basic presence control
- Camera control
- Limited to one camera
- Excluding camera presets
- Far end camera control on MultiSite (MultiSite is not supported on Quick Set C20)
- Volume control
- Microphone mute control
- EMC resilience mode
- All in one “search and dial” mechanism
- Provisioning of system settings and phonebook is supported. Provisioning of software upgrade is not supported in this release
- Password protection of the Administrator Settings

The Administrator Settings menu on the Touch controller can be password protected. This is done from a command line interface with an API (Application Programmer Interface) command. The password protection options are described in the C Series administrator guides.

NOTE: When using the touch controller most of the system configuration is done from the web interface. The web interface is described in the C Series administrator guides.

The Cisco TelePresence Touch for C Series can be connected to the codec over LAN or directly through network port 2. Direct connection applies to Codec C40, C60 and C90.
New API commands
Detailed descriptions of the API commands are found in this guide in the xConfiguration, xCommand, and xStatus sections.

<table>
<thead>
<tr>
<th>Contents</th>
<th>Introduction</th>
<th>xConfiguration</th>
<th>xCommand</th>
<th>xStatus</th>
<th>Appendices</th>
</tr>
</thead>
</table>

**New API commands**

**xConfiguration**
- xConfiguration Video Input Source [1..n] Type

**xCommand**
- xCommand CallLog Missed Reviewed
- xCommand Message TextLine Display
- xCommand Message TextLine Clear
- xCommand Message Echo
- xCommand Video Layout SetPresentationView

**xStatus**
- xStatus Camera [1..7] Capabilities Options
- xStatus Video Layout PresentationView

API commands that have changed

**xConfiguration**
- xConfiguration Provisioning Mode
  - Added argument "VCS"
- xConfiguration Video SelfviewPosition
  - Added argument "CenterRight"

**xCommand**
- xCommand Video PictureLayoutSet
  - Added argument "Speaker_Full" to the "LayoutFamily" parameter

API commands that are removed

**xConfiguration**
- xConfiguration SystemUnit Type

Experimental settings
The Experimental settings are beta settings. These settings can be used 'as is', and are not fully documented.

**NOTE:** The Experimental settings are likely to change.

New commands
- xConfiguration Experimental NetworkServices UPnP Mode
- xConfiguration Experimental NetworkServices UPnP Timeout
- xConfiguration Experimental SystemUnit MenuType

Other API commands

New commands
- SystemTools Pairing Unpair
  - The pairing command handles association with Cisco TelePresence Touch controller.
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 is 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 password 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 password 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

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)**

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

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

<table>
<thead>
<tr>
<th>DCE 9 pin</th>
<th>Direction</th>
<th>PC DTE, 9 pin</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 CD</td>
<td>--&gt;</td>
<td>1 CD</td>
</tr>
<tr>
<td>2 RD</td>
<td>--&gt;</td>
<td>2 RD</td>
</tr>
<tr>
<td>3 TD</td>
<td>--&gt;</td>
<td>3 TD</td>
</tr>
<tr>
<td>4 DTR</td>
<td>&lt;--</td>
<td>4 DTR</td>
</tr>
<tr>
<td>5 GND</td>
<td>&lt;--</td>
<td>5 GND</td>
</tr>
<tr>
<td>6 DSR</td>
<td>--&gt;</td>
<td>6 DSR</td>
</tr>
<tr>
<td>7 RTS</td>
<td>&lt;--</td>
<td>7 RTS</td>
</tr>
<tr>
<td>8 CTS</td>
<td>--&gt;</td>
<td>8 CTS</td>
</tr>
<tr>
<td>9 RI</td>
<td>--&gt;</td>
<td>9 RI</td>
</tr>
</tbody>
</table>
**Value types and formats**

The system supports the following value types:

- Integer values
- Literal values
- String values
- E164 string values (strings only containing digits, '#' and '*')
- IPv4 Address values
- IPv6 Address values*
- IPv4 or *IPv6 Address values

**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 `y` characters.
  `<S: 0, 49>`

- **IPv4 Address values:** `<IPAddr>`
  Defines that the input must be an IPv4 address.
  `<IPAddr>`

- **IPv6 Address values:** `<IPv6Addr: x, y>`
  Defines that the input must be an IPv6 address with minimum length `x` and maximum length `y`.
  `<IPv6Addr: 0, 43>`

- **IPv4 or *IPv6 Address values:** `<IPv4v6Addr: x, y>`
  Defines that the input must be an IPv4 or IPv6 address with minimum length `x` and maximum length `y`.
  `<IPv4v6Addr: 0, 43>`

---

* Not supported in this version
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 other commands

The other user commands are described in the following pages.
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 type commands are either supplied or read by the user.

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. Command type commands are supplied by the user.

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

A Command type 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 xFeedback command is 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.

More on this can be found in xfeedback.
- xFeedback
- xEvent

Preferences type command
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.

More on this can be found in xpreferences.
- xPreferences
### About xConfiguration

The xConfiguration type commands define the system settings and are either supplied or read by the user. The xConfigurations commands are organized in a hierarchical 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`

<table>
<thead>
<tr>
<th>The type</th>
<th>The path</th>
<th>The value</th>
</tr>
</thead>
<tbody>
<tr>
<td>xConfiguration</td>
<td>Audio Input HDMI 1</td>
<td>Mode: On</td>
</tr>
</tbody>
</table>

#### xConfiguration ?

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

#### xConfiguration ??

`*h xConfiguration Audio Volume: <0..100>`

Sets the volume level (0-100) on the loudspeaker output in steps of 0.5 dB from -34.5 dB to 15 dB. Volume 0 = Off. The volume level bar which is displayed on screen, when using the remote control, goes from 0 to 20. Range: The volume level goes from 0 to 100. Volume level equals Audio gain value. 0 equals 0 dB, 1 equals -34.5 dB, 70 equals 0.0 dB, 100 equals 15.0 dB.

`*h xConfiguration Audio Input Microphone [1..8] Type: <Microphone/Line>`

The microphone inputs are intended for electret type microphones. The microphone inputs are balanced with 48 V phantom power. The microphone input can be set to line or microphone mode. Addresses the specific microphone. Microphone: Phantom voltage and pre-amplification is On. Line: Select Line when you have a standard balanced line input. The phantom voltage and pre-amplification is Off.

`*h xConfiguration Audio Input Microphone [1..8] Mode: <On/Off>`

By default, all inputs are enabled. Just plug in an audio source and it is active. Audio inputs that are On will automatically be mixed. Unconnected inputs will automatically be muted. Addresses the specific microphone. On: Turns the microphone On. Off: Connected but unused inputs should be set to Off to prevent audio/noise from the inputs.

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**
  - Help text for this 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.

---

### Example with xConfiguration Help:

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

```
xConfiguration <address expression> ?
```

Example:
```
xConfiguration H323 Profile 1 Gatekeeper Discovery ?
```
```
xConfiguration H323 Profile 1 Gatekeeper Discovery: <Manual/Auto>
OK
```

### 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>
```

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

### 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>
```

Example:
```
xConfiguration H323 Profile 1 Gatekeeper Discovery: Auto
```
```
** end
```
About xCommand

The xCommand type commands instruct 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 ??

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 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**

- `*h` is used when returning the result of a help query

**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 ?
  *h xCommand Dial
  Number(r): <S: 0, 255>
  Protocol: <H323/Sip>
  CallRate: <64..6000>
  CallType: <Audio/Video>
  ForceNewConference: <False/True>
  ConferenceId: <0..65534>
  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
  *r/end
  OK
```
About xStatus commands

The xStatus type commands return 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

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
Query 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.

Example 1: Query all ongoing Call information:

```
xstatus call
*s Call 3 Status: Connected
*s Call 3 Direction: Outgoing
*s Call 3 Protocol: "h323"
*s Call 3 RemoteNumber: "firstname.lastname@company.com"
*s Call 3 CallbackNumber: "h323: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 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: "h323"
```

OK
About xHistory command

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

Status operations and the 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: Outgoing
*h xHistory CallLogs Call 1 CallType: Video
*h xHistory CallLogs Call 1 RemoteNumber: ”h323:firstname.lastname.office@company.com”
*h xHistory CallLogs Call 1 CallbackNumber: ”h323:firstname.lastname.office@company.com”
*h xHistory CallLogs Call 1 DisplayName: ”firstname.lastname@company.com”
*h xHistory CallLogs Call 1 CallRate: 768
*h xHistory CallLogs Call 1 DisconnectCauseValue: 1
*h xHistory CallLogs Call 1 DisconnectCause: ”MC:Normal”
*h xHistory CallLogs Call 1 StartTime: ”2010/04/14 11:04:14”
*h xHistory CallLogs Call 1 Duration: 164
*h xHistory CallLogs Call 1 Encryption: ”None”
...
*h xHistory CallLogs Recent 6 CounterMissed: 0
*h xHistory CallLogs Recent 7 CounterMissed: 0
...
*h xHistory CallLogs Outgoing 29 Counter: 1
*h xHistory CallLogs Outgoing 30 Counter: 1
...
*h xHistory CallLogs Missed 50 Counter: 2
** end
About xEvent command

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

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.

Example with xFeedback

```
xFeedback ?
xFeedback help:
xFeedback Register XPathExpression - Registers feedback on expression XPathExpression
xFeedback Deregister XPathExpression - Deregisters feedback if registered on XPathExpression
xFeedback List - Generate list of currently registered XPathExpressions
xFeedback Help - Display this help text
```

```
xFeedback register Status/Audio
xFeedback register Configuration/Video
xFeedback register Event

xFeedback list
xFeedback deregister Event
xFeedback list
```
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`

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.

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

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.

 proximité The encryption method used, which can be ssh-rsa or ssh-dss.

 `<key>`: 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>`: The id 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, defined by the name.

 `<name>`: The name of the license file.

**systemtools network ping <hostname>**
Network debug commands.

 `<hostname>`: The IP address or URL of the host.

**systemtools network traceroute <hostname>**
Network debug commands.

 `<hostname>`: The IP address or URL of the host.

**systemtools pairing unpair**
Remove association with Cisco TelePresence Touch controller.

**systemtools passwd**
Change the password for the logged in user.

**systemtools pki list**
Lists the codec certificate and CA list if they exist.

**systemtools pki delete <cert-name>**
Delete the codec certificate and CA list if they exist.

 `<cert-name>`: The name of the certificate.

**systemtools 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 or view password and PIN polices enforced on the codec.

### 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.

<table>
<thead>
<tr>
<th>Policy</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max failed login attempts [0]</td>
<td>Number of failed logins until a user is set inactive.</td>
</tr>
<tr>
<td>Suspend-time after max failed login attempts (minutes) [0]</td>
<td>Number of minutes the user is set inactive after maximum failed login attempts have been exceeded.</td>
</tr>
<tr>
<td>Max simultaneous sessions total [0]</td>
<td>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.</td>
</tr>
<tr>
<td>Max simultaneous sessions per user [0]</td>
<td>Maximum number of simultaneous sessions per user.</td>
</tr>
<tr>
<td>Number of passwords to remember [0]</td>
<td>Number of previous passwords that the new password must differ from.</td>
</tr>
<tr>
<td>Number of PINs to remember [0]</td>
<td>Number of previous PINs that the new PIN must differ from.</td>
</tr>
<tr>
<td>Maximum time between password renewals (days) [0]</td>
<td>If the user has not changed the password within the renewal time the user will be set inactive.</td>
</tr>
<tr>
<td>Minimum time between password renewals (hours) [0]</td>
<td>If the user has not changed the password within the renewal time the user will be set inactive.</td>
</tr>
<tr>
<td>Maximum time between PIN renewals (days) [0]</td>
<td>The user can only change PIN once within this limit.</td>
</tr>
<tr>
<td>Minimum time between PIN renewals (hours) [0]</td>
<td>The user can only change PIN once within this limit.</td>
</tr>
<tr>
<td>Max consecutive equal digits in PINs [0]</td>
<td>Maximum consecutive equal digits in PINs.</td>
</tr>
<tr>
<td>Minimum number of digits in PINs [0]</td>
<td>Minimum number of digits in PINs.</td>
</tr>
<tr>
<td>Maximum number of digits in PINs [0]</td>
<td>Maximum number of digits in PINs.</td>
</tr>
<tr>
<td>Max consecutive identical characters in passwords [0]</td>
<td>Maximum consecutive identical characters in passwords.</td>
</tr>
<tr>
<td>Minimum number of characters in passwords [0]</td>
<td>Minimum number of characters in passwords.</td>
</tr>
<tr>
<td>Maximum number of characters in passwords [0]</td>
<td>Maximum number of characters in passwords.</td>
</tr>
<tr>
<td>Minimum number of lower-case letters in passwords [0]</td>
<td>Minimum number of lower-case letters in passwords.</td>
</tr>
<tr>
<td>Minimum number of upper-case letters in passwords [0]</td>
<td>Minimum number of upper-case letters in passwords.</td>
</tr>
<tr>
<td>Minimum number of numerical characters in passwords [0]</td>
<td>Minimum number of numerical characters in passwords.</td>
</tr>
<tr>
<td>Minimum number of special characters in passwords [0]</td>
<td>Minimum number of special characters in passwords.</td>
</tr>
<tr>
<td>Minimum number of character groups in passwords [0]</td>
<td>Minimum number of character groups in passwords.</td>
</tr>
<tr>
<td>Minimum number of character changed from previous password [0]</td>
<td>Minimum number of character changed from previous password.</td>
</tr>
</tbody>
</table>
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-urlencoded 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 show either the default value or an example of a value.

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

The Audio settings

xConfiguration Audio Input HDMI [2] Level
Define the audio level of the HDMI input connector, in steps of 1dB.
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 Input HDMI 2 Level: 0

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] VideoAssociation MuteOnInactiveVideo
Enable association of a video source to a 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 three video input sources.
Example: xConfiguration Audio Input HDMI 2 VideoAssociation VideoInputSource: 1
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 Line [1..2] Equalizer ID
Select the audio input line equalizer ID.

Requires user role: ADMIN
Value space: <1..8>
  Range: Select Equalizer ID 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 VideoInputSource
Select the associated video input source.

Requires user role: ADMIN
Value space: <1/2/3>
  Range: Select one of the three video input sources.

Example: xConfiguration Audio Input Line 1 VideoAssociation VideoInputSource: 1

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] 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 Line 1 Channel: Left

xConfiguration Audio Input Line [1..2] Level
Define the audio level of the Line input connector, in steps of 1dB.
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 24dB.

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 compensates 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..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..14>
  Range: Select Equalizer ID 1 to 14.

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 three 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 1dB. 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 24dB.

Example: xConfiguration Audio Input Microphone 1 Level: 14
**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 1dB.

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:** <-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 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 Equalizer ID 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 1db.

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:** <-24..0>

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

**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 1 Type: Analog

**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 key tone on the remote control is switched off.
- **Example:** xConfiguration Audio SoundsAndAlerts KeyTones Mode: Off

**xConfiguration Audio SoundsAndAlerts RingTone**

Selects the ringtone for incoming calls.

- **Requires user role:** USER
- **Value space:** <Marbles/IceCrystals/Polaris/Alert/Discrete/Fantasy/Jazz/Nordic/Echo/Rhythmic>
  - **Range:** Select a tone from the list of ringtones.
- **Example:** xConfiguration Audio SoundsAndAlerts RingTone: Jazz

**xConfiguration Audio SoundsAndAlerts RingVolume**

Sets the ring tone volume for an incoming call. The value goes in steps of 5 from 0 to 100 (from -34.5dB to 15dB). Volume 0 = Off.

- **Requires user role:** USER
- **Value space:** <0..100>
  - **Range:** Select a value from 0 to 100.
- **Example:** xConfiguration Audio SoundsAndAlerts RingVolume: 50

**xConfiguration Audio Volume**

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

- **Requires user role:** USER
- **Value space:** <0..100>
  - **Range:** Select a value from 0 to 100.
- **Example:** xConfiguration Audio Volume: 70
The Cameras settings

**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/60Hz**: Set to 50Hz or 60Hz.

*Example:* xConfiguration Cameras PowerLine Frequency: Auto

**xConfiguration Cameras Camera [1..7] Backlight**
The backlight functionality compensates for lights shining directly at the camera (usually the sun entering the window) to avoid a too dark image from the room.

- Requires user role: **ADMIN**
- Value space: `<On/Off>`
  - **On**: Turn on the camera backlight.
  - **Off**: Turn off the camera backlight.

*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. Use this setting with cameras that can be mounted upside down, and that can auto detect that the camera is mounted upside down.
  - **On**: When set to On the video on screen is flipped. This setting is used with cameras that can be mounted upside down, but cannot auto detect that the camera is mounted upside down.
  - **Off**: Set to Off to 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**: When set to Auto the focus will be updated throughout the call. When moving the camera, the system will use auto focus for a few seconds to set the right focus of the new camera position. After a few seconds 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 supports 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: When set to Auto, the camera will continuously adjust the whitebalance depending on the camera view.
  - Manual: Set to Manual to enable 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 supports DHCP. The Cisco TelePresence PrecisionHD 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 command “xStatus Camera” for result.
  - Off: Set to Off will disable DHCP in the camera. NOTE: When camera is not connected to a LAN, this setting should be applied.
- Example: `xConfiguration Cameras Camera 1 DHCP: Off`
The Conference settings

**xConfiguration Conference [1..1] IncomingMultisiteCall Mode**
Set the incoming Multisite call mode.

Requires user role: ADMIN

Value space: <Allow/Deny>
- **Allow**: Accept incoming calls at an already active call/conference. The incoming call will be added to the MCU conference.
- **Deny**: The system will not accept incoming calls when you are in a call. The calling side will receive a busy signal.

Example: `xConfiguration Conference 1 IncomingMultisiteCall Mode: Allow`

**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**
Determine if the microphones should be unmuted automatically when all calls are disconnected. In a meeting room or other shared resource 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: 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>
- **On**: All incoming calls will be rejected, with no alert. The calling side will receive a busy signal when trying to call the codec. A message will display on screen, telling that Do not disturb is turned on, together with an option to turn off the Do not disturb. When turning off the Do not disturb mode you will see a list of the calls that have been rejected.
- **Off**: The incoming calls will be alerted.

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**: Set to On when you want 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**: When set to Off the far end can not access any of the features above on your system.

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: 64-6000 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 from 64 to 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 from 64 to 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 packet loss 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 packet loss resilience.
- **Off:** Disable the packet loss resilience.

**Example:** xConfiguration Conference 1 PacketLossResilience Mode: On

The GPIO settings

### 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 (+12V). When activated as output, they are set to 0V. To activate them as input, they must be pulled down to 0V.

**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 +12V or 0V, 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 (0V), the microphones will be muted. When deactivated (+12V), the microphones are unmuted.

**Example:** xConfiguration GPIO Pin 1 Mode: InputNoAction
## The H323 settings

### 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.

<table>
<thead>
<tr>
<th>Requires user role:</th>
<th>ADMIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value space:</td>
<td>&lt;Auto/On/Off&gt;</td>
</tr>
<tr>
<td>Auto:</td>
<td>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.</td>
</tr>
<tr>
<td>On:</td>
<td>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”.</td>
</tr>
<tr>
<td>Off:</td>
<td>The system will signal the real IP Address.</td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Requires user role:</th>
<th>ADMIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value space:</td>
<td>&lt;S: 0, 64&gt;</td>
</tr>
<tr>
<td>Format:</td>
<td>String with a maximum of 64 characters.</td>
</tr>
</tbody>
</table>

**Example:** `xConfiguration H323 NAT Address: ""`

### xConfiguration H323 Profile [1..1] Authentication Mode
Set the authentication mode for the H.323 profile.

<table>
<thead>
<tr>
<th>Requires user role:</th>
<th>ADMIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value space:</td>
<td>&lt;On/Off&gt;</td>
</tr>
<tr>
<td>On:</td>
<td>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.</td>
</tr>
<tr>
<td>Off:</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Requires user role:</th>
<th>ADMIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value space:</td>
<td>&lt;S: 0, 50&gt;</td>
</tr>
<tr>
<td>Format:</td>
<td>String with a maximum of 50 characters.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Requires user role:</th>
<th>ADMIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value space:</td>
<td>&lt;S: 0, 50&gt;</td>
</tr>
<tr>
<td>Format:</td>
<td>String with a maximum of 50 characters.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Requires user role:</th>
<th>ADMIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value space:</td>
<td>&lt;Direct/Gatekeeper&gt;</td>
</tr>
<tr>
<td>Direct:</td>
<td>An IP-address must be used when dialling in order to make the H323 call.</td>
</tr>
<tr>
<td>Gatekeeper:</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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"
The Network settings

**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*: Compact 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 IP 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 IP 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 IP 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.

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.

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).
Example: xConfiguration Network 1 IPv6 Assignment: Autoconf

xConfiguration Network [1..1] IPv6 DHCP Options
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.
Diff serv: When you set the QoS Mode to Diff serv you must configure the Diff serv sub menu settings (Audio, Data, Signalling and Video).
Example: xConfiguration Network 1 QoS Mode: diffserv

xConfiguration Network [1..1] QoS Diff serv Audio
The Diff serv 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 Diff serv.

Requires user role: ADMIN
Value space: <0..63>
Audio: A recommended value is Diff serv 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 Diff serv Audio: 0

xConfiguration Network [1..1] QoS Diff serv Data
The Diff serv 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 Diff serv.

Requires user role: ADMIN
Value space: <0..63>
Data: A recommended value is Diff serv 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 Diff serv Data: 0

xConfiguration Network [1..1] QoS Diff serv Signalling
The Diff serv 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 Diff serv.

Requires user role: ADMIN
Value space: <0..63>
Signalling: A recommended value is Diff serv 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 Diff serv 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 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] MTU**
Set the Ethernet MTU (Maximum Transmission Unit).

Requires user role: ADMIN

Value space: <400..1500>
  
  **Range:** Select a value from 400 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 10Mbps half-duplex.
  
  **10full:** Force link to 10Mbps full-duplex.
  
  **100half:** Force link to 100Mbps half-duplex.
  
  **100full:** Force link to 100Mbps full-duplex.
  
  **1000full:** Force link to 1Gbps 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 20Mbps. 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] VLAN Voice Mode**
Set the VLAN voice mode.

Requires user role: ADMIN

Value space: <Tagged/Untagged>
  
  **Tagged:** The voice packets in the VLAN network are tagged with VlanId and Priority.
  
  **Untagged:** The voice packets in the VLAN network are untagged.

Example: xConfiguration Network 1 VLAN Voice Mode: Untagged

**xConfiguration Network [1..1] VLAN Voice VlanId**
Set the VLAN voice ID.

Requires user role: ADMIN

Value space: <0..4096>
  
  **Range:** Select a value from 0 to 4096.

Example: xConfiguration Network 1 VLAN Voice VlanId: 0

**xConfiguration Network [1..1] VLAN Voice Priority**
Set the VLAN voice priority.

Requires user role: ADMIN

Value space: <0..7>
  
  **Range:** Select a value from 0 to 7.

Example: xConfiguration Network 1 VLAN Voice Priority: 0

**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 addresses or IP range.

Example: xConfiguration Network 1 RemoteAccess Allow: "192.168.1.231, 192.168.1.182"
The NetworkPort settings


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 settings

**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: Codec C60/C40 must be running TC4.0 (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. 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 list of trusted CA's 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 list of trusted CA's are uploaded to the system in advance.
Off: Do not verify client certificates.

Example: xConfiguration NetworkServices HTTPS VerifyClientCertificate: Off

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

Requires user role: ADMIN
Value space: <Auto/Manual>
Auto: The system will use the NTP server, by which address is supplied from the DHCP server in the network. If no DHCP server is used, or the DHCP server does not provide the system with a NTP server address, the system will use the static defined NTP server address specified by the user.
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.tandberg.pool.ntp.org"

xConfiguration NetworkServices SIP Mode
Determine whether the system should be able to place and receive SIP calls. 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

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

Enter the address of up to three SNMP Managers. All traps will then be sent to the hosts listed. The system's SNMP Agent (in the codec) responds to requests from SNMP Managers (a PC program etc.). SNMP Traps are generated by the SNMP Agent to inform the SNMP Manager about important events. Can be used to send event created messages to the SNMP agent about different events like: system reboot, system dialling, system disconnecting, MCU call, packet loss etc. Traps can be sent to multiple SNMP Trap Hosts.

**Requires user role:** ADMIN

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

**Format:** String with a maximum of 64 characters.

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

## 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"

## 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: ""

## 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: ""

## 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

## 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 settings

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>
  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.

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.

The Provisioning settings

**xConfiguration Provisioning Mode**
Provides the possibility of managing the codec (endpoint) by using an external manager/management system.

**Requires user role:** ADMIN

**Value space:** `<Off/TMS/VCS/Callway>`
- **Off:** The system will not try to register to any management system.
- **TMS:** If set to TMS (Cisco TelePresence Management System) the system will try to register with a TMS server. Contact your Cisco representative for more information.
- **VCS:** If set to VCS (Cisco TelePresence Video Communication Server) the system will try to register with a VCS. Contact your Cisco representative for more information.
- **Callway:** If set to Callway the system will try to register with the Callway subscription provider. Contact your Callway provider for more information.

**Example:** *xConfiguration Provisioning Mode: TMS*

**xConfiguration Provisioning ExternalManager Address**
Enter the IP Address to the External Manager/Management system. If an External Manager address and a path is configured, the system will post an HTTP message to this address when starting up. When receiving this HTTP posting the External Manager (typically a management system) can return configurations/commands to the unit as a result. If the DHCP Option 242 is returned in the DHCP response from the DHCP server the system will interpret this as the External Manager address to use.

**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 uninvalid IP addresses (192.0.1234.0) will be rejected.

**Example:** *xConfiguration Provisioning ExternalManager Address: ""*

**xConfiguration Provisioning ExternalManager Domain**
Enter the SIP domain for the 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"*

**xConfiguration Provisioning ExternalManager Path**
Set the path to the External Manager/Management system. If an External Manager address and a path is configured, the system will post an HTTP message to this address when starting up. When receiving this HTTP posting the External Manager (typically a management system) can return configurations/commands to the unit as a result. If the DHCP Option 242 is returned in the DHCP response from the DHCP server the system will interpret this as the External Manager address to use.

**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 Protocol**
Determine whether or not to use secure management.

**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 LoginName**
Enter the user id provided by the provisioning server. This is the user name part of the credentials used to authenticate towards the HTTP server when using HTTP provisioning.

**Requires user role:** ADMIN

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

**Example:** *xConfiguration Provisioning LoginName: ""*

**xConfiguration Provisioning Password**
Enter the password provided by the provisioning server. This is the password part of the credentials used to authenticate towards the HTTP server when using HTTP provisioning.

**Requires user role:** ADMIN

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

**Example:** *xConfiguration Provisioning Password: ""*
### The Security settings

**xConfiguration Security Audit Server Address**

Enter the external/global IP-address to the audit syslog server.

- **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. A user with AUDIT rights is required to change this setting.

- **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. A user with AUDIT rights is required to change this setting.

- **Requires user role:** AUDIT
- **Value space:** `<Halt/Ignore>`
  - **Halt:** If the connection to the syslog server is lost for more than a few seconds, the system will reboot and try to establish connection. If connection is restored, the audit logs are respoled to the syslog server, and the system starts up again.
  - **Ignore:** The system will continue its normal operation, and rotate internal logs when full. When connection is restored it will again sends 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. A user with AUDIT rights is required to change this setting.

Requires user role: AUDIT

Value space: <Off/Internal/External/ExternalSecure>
- Off: No audit logging is performed.
- Internal: The system records the audit logs to internal logs, and rotates logs when they are full.
- External: The system sends the audit logs to an external audit server.
- ExternalSecure: The system sends the audit logs to an external audit server that is verified by the Audit CA list.

Example: xConfiguration Security Audit Logging Mode: 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: AUDIT

Value space: <0..10000>
- Range: Select a value from 0 to 10000 seconds. 0 means the that inactivity will not enforce automatically logout.

Example: xConfiguration Security Session InactivityTimeout: 0

The SerialPort settings

xConfiguration SerialPort Mode

Set the COM 1 serial port to be enabled/disabled.

Requires user role: ADMIN

Value space: <On/Off>
- On: Enable the COM 1 serial port.
- Off: Disable the COM 1 serial port.

Example: xConfiguration SerialPort Mode: On

xConfiguration SerialPort BaudRate

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

Connection parameters for the COM port: 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 COM port at the codec.

Requires user role: ADMIN

Value space: <On/Off>
- On: Login is required when connecting to the codec through COM port.
- Off: The user can access the codec through COM port without any login.

Example: xConfiguration SerialPort LoginRequired: On
The SIP settings

**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, 50>
- Format: String with a maximum of 50 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, 50>
- Format: String with a maximum of 50 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 using the web interface. 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 using the web interface.

- Requires user role: ADMIN
- Value space: <On/Off>
  - On: Set to On to verify TLS connections. Only TLS connections to servers, whom 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**

- 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>
  - Manual: When Manual is selected, the manually configured SIP Proxy address will be used.
  - Auto: When Auto is selected, the SIP Proxy address is obtained using Dynamic Host Configuration Protocol (DHCP).
- 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/Experimental/Siemens>

- **Standard**: Should be used when registering to standard SIP proxy like OpenSIP.
- **Alcatel**: Must be used when registering to an Alcatel-Lucent OmniPCX Enterprise R7 or later.
- **Avaya**: Must be used when registering to an Avaya Communication Manager.
- **Cisco**: Must be used when registering to a Cisco CallManager version 5 or later.
- **Microsoft**: Must be used when registering to a Microsoft LCS or OCS server.
- **Nortel**: Must be used when registering to a Nortel MCS 5100 or MCS 5200 PBX.
- **Experimental**: Can be used if the above modes are not working. NOTE: This mode is for testing purposes only.

Example: xConfiguration SIP Profile 1 Type: Standard

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"

The Standby settings

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 settings

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/Danish/Dutch/Finnish/ French/German/Italian/Japanese/Korean/Polish/PortugueseBrazilian/ Russian/Spanish/SpanishLatin/Swedish/Turkish>
Example: xConfiguration SystemUnit MenuLanguage: English

xConfiguration SystemUnit ContactInfo Type
Describes what 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 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 menus on screen 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 Mode
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 Mode: Auto

The Time settings

xConfiguration Time DateFormat
Set the date format.

Requires user role: USER
Value space: <DD _ MM _ YY/MM _ DD/YY _ MM _ DD>
- DD_MM_YY: The date January 30th 2010 will be displayed: 30.01.10
- MM_DD_YY: The date January 30th 2010 will be displayed: 01.30.10
- YY.MM_DD: The date January 30th 2010 will be displayed: 10.01.30
Example: xConfiguration Time DateFormat: DD _ MM _ YY

xConfiguration Time 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 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-06:00 (Chihuahua, La Paz, Mazatlan)/GMT-05:00 (Central America)/GMT-05:00 (Chicago, Dallas, Houston, Indianapolis, Mexico City, Moncton)/GMT-04:00 (Atlantic Time (Canada))/GMT-04:00 (Buenos Aires, Georgetown)/GMT-03:00 (Bermuda)/GMT-03:00 (Brasilia)/GMT-03:00 (Mid-Atlantic)/GMT-02:00 (Azores)/GMT-02:00 (Casablanca, Monrovia)/GMT-02:00 (Cape Verde Is.)/GMT-01:00 (Bengaluru, Colombo, Dhaka, Kolkata, Sri Jayawardenepura Kotte)/GMT-01:00 (Bangkok, Chiang Mai, Hanoi, Jakarta, Kuala Lumpur, Singapore)/GMT-01:00 (Bahrain)/GMT-01:00 (Berlin, Bern, Rome, Stockholm, Vienna)/GMT-01:00 (Amsterdam, Brussels, Copenhagen, Madrid, Paris)/GMT-01:00 (Bratislava, Budapest, Ljubljana, Prague)/GMT-01:00 (Harare, Pretoria)/GMT-01:00 (Jerusalem)/GMT-00:00 (Athens, Istanbul, Minsk)/GMT-00:00 (Belgrade, Bratislava, Budapest, Ljubljana, Prague)/GMT+00:00 (Aeroparque, Buenos Aires)/GMT+00:00 (Abu Dhabi, Muscat)/GMT+00:00 (Beijing, Chongqing, Hong Kong, Urumqi)/GMT+00:00 (Brussels, Copenhagen, Madrid, Paris)/GMT+00:00 (Bucharest)/GMT+00:00 (Cairo)/GMT+00:00 (Casablanca, Monrovia)/GMT+00:00 (Islamabad, Karachi, Tashkent)/GMT+00:00 (Athens, Brussels, Copenhagen, Madrid, Paris)/GMT+00:00 (Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna)/GMT+00:00 (Warsaw, Nairobi)/GMT+00:00 (Rangoon)/GMT+00:00 (Skopje, Warsaw, Zagreb)/GMT+00:00 (Brussels, Copenhagen, Madrid, Paris)/GMT+00:00 (Belgrade, Bratislava, Budapest, Ljubljana, Prague)/GMT+00:00 (Harare, Pretoria)/GMT+00:00 (Jerusalem)/GMT+00:00 (Athens, Istanbul, Minsk)/GMT+00:00 (Belgrade, Bratislava, Budapest, Ljubljana, Prague)/GMT+00:00 (Harare, Pretoria)/GMT+00:00 (Jerusalem)/GMT+01:00 (Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna)/GMT+01:00 (Brussels, Copenhagen, Madrid, Paris)/GMT+01:00 (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+03:00 (Tehran)/GMT+04:00 (Abu Dhabi, Muscat)/GMT+04:00 (Baku, Tbilisi, Yerevan)/GMT+04:30 (Kabul)/GMT+05:00 (Chennai, Kolkata, Mumbai, New Delhi)/GMT+05:30 (Irkutsk, Ulaan Batar)/GMT+05:30 (Osaka, Sapporo, Tokyo)/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 Batar)/GMT+09:00 (Osaka, Sapporo, Tokyo)/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., New Zealand)/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)"

The Video settings

xConfiguration Video DefaultPresentationSource

Define which video input source shall be used as the default presentation source (e.g. 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..3>

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 Input Source [1..3] Name

Enter a name for the video input source 1 to 3.

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:** <camera/PC/DVD/document_camera/other>
  - **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.
- **Other:** Select Other when other equipment is connected to the selected video input.
- **Example:** xConfiguration Video Input Source 1 Type: Camera

**xConfiguration Video Input Source [1..5] 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..5>
- **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] CameraControl Mode**
Set the camera control mode for the camera associated with the video source 1 to 3.
- **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] OptimalDefinition Profile**

Adjust how rapidly the system will increase the transmitted resolution when increasing the bandwidth. NOTE: Requires that the Video Input Source Quality is set to Motion.

Normal: Use this setting for normal to poorly lit environment. If the source is a camera with 1920x1080p60, the system will transmit 1920x720p60 at about 2.2Mb/sec and above with this setting set to normal.

Medium: Requires better than normal and consistent lighting and good quality video inputs. If the source is a camera with 1920x1080p60, the system will transmit 1920x720p60 at about 1.4Mb/sec and above with this setting set to medium.

High: Requires good lighting conditions for a good overall experience and good quality video inputs. If the source is a camera with 1920x1080p60, the system will transmit 1920x720p60 at about 1.1Mb/sec and above with this setting set to high.

Requires user role: ADMIN

Value space: <Normal/Medium/High>

Ref: Table 1 and Table 2.

**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 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`

---

**Table 1: Optimal definition, for systems supporting 1080p**

<table>
<thead>
<tr>
<th></th>
<th><code>w288p30</code></th>
<th><code>w448p30</code></th>
<th><code>w576p30</code></th>
<th><code>720p30</code></th>
<th><code>1080p30</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>256 kbit/s</td>
<td>512 kbit/s</td>
<td>768 kbit/s</td>
<td>1152 kbit/s</td>
<td>2560 kbit/s</td>
</tr>
<tr>
<td>Medium</td>
<td>128 kbit/s</td>
<td>384 kbit/s</td>
<td>512 kbit/s</td>
<td>1152 kbit/s</td>
<td>1920 kbit/s</td>
</tr>
<tr>
<td>High</td>
<td>128 kbit/s</td>
<td>256 kbit/s</td>
<td>512 kbit/s</td>
<td>768 kbit/s</td>
<td>1472 kbit/s</td>
</tr>
</tbody>
</table>

**Table 2: Optimal definition, for systems supporting 720p60**

<table>
<thead>
<tr>
<th></th>
<th><code>w144p60</code></th>
<th><code>w288p60</code></th>
<th><code>w448p60</code></th>
<th><code>w576p60</code></th>
<th><code>720p60</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>128 kbit/s</td>
<td>512 kbit/s</td>
<td>1152 kbit/s</td>
<td>1472 kbit/s</td>
<td>2240 kbit/s</td>
</tr>
<tr>
<td>Medium</td>
<td>128 kbit/s</td>
<td>384 kbit/s</td>
<td>768 kbit/s</td>
<td>1152 kbit/s</td>
<td>1920 kbit/s</td>
</tr>
<tr>
<td>High</td>
<td>128 kbit/s</td>
<td>256 kbit/s</td>
<td>512 kbit/s</td>
<td>768 kbit/s</td>
<td>1152 kbit/s</td>
</tr>
</tbody>
</table>

---

**xConfiguration Video Input Source [1..3] Quality**

When encoding and transmitting video there will be a tradeoff 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 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`
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).

**Requirements:**
- 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

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.

**Requirements:**
- User role: ADMIN

**Value space:**<0..100>
- **Range:** Select a value from 0 to 100 percent.

**Example:** xConfiguration Video Layout ScaleToFrameThreshold: 5

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

**Requirements:**
- 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

Select which video layout family to be used locally.

**Requirements:**
- 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 command: xCommand Video Layout LoadDb.
- **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

Select which video layout family to be used for the remote participants.

**Requirements:**
- 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

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

**Requirements:**
- User role: USER

**Value space:**<1..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**: Show 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 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 today's 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 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 _ 60/1360 _ 768 _ 60/1366 _ 768 _ 60/1440 _ 900 _ 60/1920 _ 1080 _ 60/1280 _ 720 _ 60/1366 _ 768 _ 60/1366 _ 768 _ 60/1600 _ 1200 _ 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@60p, 1920x1080@60p, 1280x768@60p, 1360x768@60p, 1600x1200@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_640_800_600_60_640_1024_768_60_1280_1024_60_1280_720_1080_60_1280_768_60_1360_768_60_1600_1280_60_1280_720_1080_60_1280_768_60_1360_768_60_1600_1280_60_960_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@60p, 1920x1080@60p, 1280x768@60p, 1360x768@60p, 1366x768@60p, 1600x1200@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>

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: ADMIN

Value space: <On/Off>

- **On**: Set to On when you want selfview to be displayed on screen.
- **Off**: Set to Off when you do not want selfview to be displayed on screen.

Example: xConfiguration Video Selfview: On

---

**xConfiguration Video Wallpaper**

Determine if a background picture shall be displayed on screen when idle.

Requires user role: USER

Value space: <None/Growing/Summersky/Custom>

- **None**: No wallpaper will be displayed on screen.
- **Summersky, Growing**: Select one of the predefined wallpapers to be displayed on screen.

1. **On the video system**: Find the IP address of the codec. Open the menu on screen and go to Home > Settings > System information to find the IP Address.
2. **On your computer**: Open a web browser and enter the IP address of the codec. Select “Wallpaper” from the menu, browse for the file, and press the “Upload” button.
3. **On the video system**: Open the menu on screen and go to Home > Settings > Wallpaper > Custom. Give it a few seconds to display the new picture. If the picture does not show, toggle once between “None” and “Custom” wallpaper to make the change take effect.

Example: xConfiguration Video Wallpaper: Summersky
xConfiguration Video AllowWebSnapshots

Allows web snapshots to be taken from the web interface.

NOTE: This is a local setting which is available only from the On Screen Display (OSD) and when connected directly to the serial port (COM port) on the codec.

Requires user role: ADMIN

Value space: <On/Off>
- On: If set to on, a web snapshot can be generated and displayed on the web page under "Snapshot".
- Off: The generation of web snapshots is not allowed.

Example: xConfiguration Video AllowWebSnapshots: Off

The Experimental settings

The Experimental settings are beta preview features and can be used 'as is'. They are not fully documented.

NOTE: The Experimental settings are likely to change without further notice.

xConfiguration Experimental CustomSoftbuttons State [1..2] Softbutton [1..5] Type

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Requires user role: ADMIN

Value space: <NotSet/MainSource/PresentationSource/CameraPreset/Actions/SpeedDial>

Example: xConfiguration Experimental CustomSoftbuttons State 1 Softbutton 1 Type: NotSet

xConfiguration Experimental CustomSoftbuttons State [1..2] Softbutton [1..5] Value

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Requires user role: ADMIN

Value space: <S: 0, 255>

Example: xConfiguration Experimental CustomSoftbuttons State 1 Softbutton 1 Value: ""

xConfiguration Experimental SystemUnit Controller Address

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Requires user role: ADMIN

Value space: <S: 0, 255>

Example: xConfiguration Experimental SystemUnit Controller Address: ""
**xConfiguration Experimental Conference [1..1] PacketLossResilience ForwardErrorCorrection**

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Will enable ForwardErrorCorrection (RFC5109) mechanism as part of the PacketLossResilience mechanism. Default value is On.

On: Forward error correction will be used as part of the PacketLossResilience mechanism.

Off: Forward error correction will NOT be used as part of the PacketLossResilience mechanism.

Requires user role: ADMIN

Value space: <On/Off>

Example:
```
xConfiguration Experimental Conference 1 PacketLossResilience ForwardErrorCorrection: On
```

**xConfiguration Experimental Conference [1..1] PacketLossResilience RateAdaption**

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Will use the a RateAdaption algorithm adapted to the PacketLossResilience mechanism. Default value is On.

On: RateAdaption will be used as part of the PacketLossResilience mechanism.

Off: RateAdaption will NOT be used as part of the PacketLossResilience mechanism.

Requires user role: ADMIN

Value space: <On/Off>

Example:
```
xConfiguration Experimental Conference 1 PacketLossResilience RateAdaption: On
```

**xConfiguration Experimental SoftwareUpgrade Mode**

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Requires user role: ADMIN

Value space: <Auto/Manual>

Example:
```
xConfiguration Experimental SoftwareUpgrade Mode: Auto
```

**xConfiguration Experimental SoftwareUpgrade ServerAddress**

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Requires user role: ADMIN

Value space: <S: 0, 255>

Example:
```
xConfiguration Experimental SoftwareUpgrade ServerAddress: "http://csupdate.tandberg.com/getswlist.py"
```

**xConfiguration Experimental CapsetFilter**

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Requires user role: ADMIN

Value space: <S: 0, 100>

Example:
```
xConfiguration Experimental CapsetFilter: ""
```

**xConfiguration Experimental NetworkServices UPnP Mode**

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Requires user role: ADMIN

Value space: <On/Off>

Example:
```
xConfiguration Experimental NetworkServices UPnP Mode: Off
```

**xConfiguration Experimental NetworkServices UPnP Timeout**

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Requires user role: ADMIN

Value space: <0..3600>

Example:
```
xConfiguration Experimental NetworkServices UPnP Timeout: 0
```

**xConfiguration Experimental Audio Input Microphone [1..4] EchoControl HighPassFilter**

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Requires user role: ADMIN

Value space: <On/Off>

Example:
```
xConfiguration Experimental Audio Input Microphone 1 EchoControl HighPassFilter: Off
```

---

Cisco TelePresence System Codec C60/C40 Application programmer interface guide

D14642.04 Cisco TelePresence System Codec C60–C40 API Guide TC4.1 February 2011.
© 2010-2011 Cisco Systems, Inc. All rights reserved. www.cisco.com
**xConfiguration Experimental Audio Input Microphone [1..4] Channel**

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

NOTE: Codec C40 has two microphone connectors. Codec C60 has four microphone connectors. Defines whether the signal from the microphone is a mono signal or part of a multichannel signal.

Requires user role: ADMIN

Value space: <Left/Right/Mono>

- **Left**: The Audio Microphone input signal is the left channel of a stereo signal.
- **Right**: The Audio Microphone input signal is the right channel of a stereo signal.
- **Mono**: The Audio Microphone input signal is a mono signal.

Example: `xConfiguration Experimental Audio Input Microphone 1 Channel: Left`

**xConfiguration Experimental Audio Microphone Reinforcement Gain**

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Microphone reinforcement is local amplification of one or several input connectors, directly to one or several output connectors, with minimum delay.

For software version TC4.0: Microphone reinforcement should not be enabled on a codec that is also configured using the Audio Console application.

This setting configures the gain of this path, from -53 dB to +15 dB. The value -54 corresponds to mute.

Requires user role: ADMIN

Value space: <-54..15>

Example: `xConfiguration Experimental Audio Microphone Reinforcement Gain: -19`

**xConfiguration Experimental Audio Microphone Reinforcement AGC**

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Microphone reinforcement is local amplification of one or several input connectors, directly to one or several output connectors, with minimum delay.

For software version TC4.0: Microphone reinforcement should not be enabled on a codec that is also configured using the Audio Console application.

This setting controls an AGC on the mix of all input connectors attached to the Microphone reinforcement.

Requires user role: ADMIN

Value space: <On/Off>

Example: `xConfiguration Experimental Audio Microphone Reinforcement AGC: Off`

**xConfiguration Experimental Audio Microphone Reinforcement Input Microphone [1..4] Mode**

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Microphone reinforcement is local amplification of one or several input connectors, directly to one or several output connectors, with minimum delay.

For software version TC4.0: Microphone reinforcement should not be enabled on a codec that is also configured using the Audio Console application.

Configuring this setting to On for a microphone, means to attach it to the microphone reinforcement for local amplification. This is done in parallel to the microphone’s existing connections, and will not affect any of these.

Requires user role: ADMIN

Value space: <On/Off>

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

**xConfiguration Experimental Audio Microphone Reinforcement Output Line [1..2] Mode**

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Microphone reinforcement is local amplification of one or several input connectors, directly to one or several output connectors, with minimum delay.

For software version TC4.0: Microphone reinforcement should not be enabled on a codec that is also configured using the Audio Console application.

Configuring this setting to On for an output line, means to attach that output connector to the microphone reinforcement for local amplification. If the output connector already is attached to a Local Output, it will first be detached from that before being attached to the Microphone reinforcement. When this setting is On, the output connector cannot be attached to any Local Output.

Requires user role: ADMIN

Value space: <On/Off>

Example: `xConfiguration Experimental Audio Microphone Reinforcement Output Line 1 Mode: On`

**xConfiguration Experimental Audio Panning Mode**

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Requires user role: ADMIN

Value space: <Off/Auto>

Example: `xConfiguration Experimental Audio Panning Mode: Off`
xConfiguration Experimental Audio Panning MaxAngle

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Requires user role: ADMIN

Value space: <0..90>

Example: xConfiguration Experimental Audio Panning MaxAngle: 0

xConfiguration Experimental Audio Panning MonitorLeft

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Requires user role: ADMIN

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

Example: xConfiguration Experimental Audio Panning MonitorLeft: 1

xConfiguration Experimental Audio Panning MonitorRight

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Requires user role: ADMIN

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

Example: xConfiguration Experimental Audio Panning MonitorRight: 1

xConfiguration Experimental SystemUnit MenuType

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Requires user role: ADMIN

Value space: <Indicators/Full>

Example: xConfiguration Experimental SystemUnit MenuType: Full
Chapter 4
Description of the xCommand commands
xCommands with parameters
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"
   ** end

xCommand Audio Equalizer List
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 at the Developer Zone web page. Go to: http://

Read more about the Application Programmer Interface (API) in the API Guide for the codec. Go to:
http://www.cisco.com/go/telepresence/docs and navigate to TelePresence Solutions Endpoints >
TelePresence Integrator Products and select the Codec C Series model.

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"

** 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

Creates 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.

Read more about the audio API in the API Guide for the codec. The API guide is found at http://www.cisco.com/go/telepresence/docs.

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:
```plaintext
xCommand Audio LocalOutput Update OutputId: 5 Name: "Loudspeaker"
Loudspeaker: On Channels: 2
OK
*r AudioOutputGroupUpdateResult (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:
```plaintext
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 a connector.

Requires user role: USER

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

Example:
```plaintext
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: USER

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 RemoteInput Update

Update the automatic gain control setting of the remote input, given by the input ID.

InputId: A unique identifier for the remote input.
AGC: Set the Automatic Gain Control.

Requires user role: USER

Parameters:
- InputId(r): <0..65534>
- AGC(r): <On/Off>

Example:
xCommand Audio RemoteInput Update InputId:9 AGC: Off
OK
*r AudioRemoteInputGroupUpdateResult (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:** USER

**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. Read more about the Application Programmer Interface (API) in the API Guide for the codec. Go to: http://www.cisco.com/go/telepresence/docs.

**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
OK
*r AudioSoundPlayResult (status=OK):
** end```
The Boot command

**xCommand Boot**

Reboot system.

Requires user role: USER

Example:

```
xCommand Boot
*r BootResult (status=OK):
** end
OK
```

CUIL reboot request, restarting
Connection closed by foreign host.

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
```
### 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:**

```plaintext
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:**

```plaintext
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:**

```plaintext
xCommand Call Disconnect CallId:17
OK
*r DisconnectCallResult (status=OK):
** end
```

#### xCommand Call DisconnectAll

Disconnect all active calls.

**Requires user role:** USER

**Example:**

```plaintext
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: <0..65534>

**Example:**

```plaintext
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

**Parameters:**
- No parameters.

**Example:**

```plaintext
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

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 LogTag:202
  *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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogTag</td>
<td>LogTag: &lt;0..2147483647&gt;</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogTag</td>
<td>LogTag: &lt;0..2147483647&gt;</td>
</tr>
</tbody>
</table>

Example:

```
xCommand CallLog Missed Delete LogTag:119
  *r DeleteResult (status=OK):
  ** end
```

**xCommand CallLog Missed Reviewed**

Review the call log of missed calls. If a logtag is given as argument, that specific call is reviews from the log. If no logtag is given, the complete missed calls log will be reviewed. The LogTag values for missed calls are found by issuing the xHistory CallLog Missed command.

Requires user role: USER

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogTag</td>
<td>LogTag: &lt;0..2147483647&gt;</td>
</tr>
</tbody>
</table>

Example:

```
xCommand CallLog Missed Reviewed LogTag:119
  *r DeleteResult (status=OK):
  ** end
```

---

**The CamCtrlPip command**

**xCommand CamCtrlPip**

Show or hide the camera selfview in a small window (picture in picture).

Requires user role: USER

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode(r)</td>
<td>Mode(r): &lt;On/Off&gt;</td>
</tr>
</tbody>
</table>

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 to the default 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. NOTE: You must run a stop command to stop the camera, see the example below.
CameraId: Give the camera id.
Pan: Move the camera to the Left or Right, followed by Stop.
PanSpeed: Set the pan speed.
Tilt: Move the camera Up or Down, followed by Stop.
TiltSpeed: Set the tilt speed.
Zoom: Zoom the camera In or Out, followed by Stop.
ZoomSpeed: Set the zoom speed.
Focus: Focus the camera Far or Near, followed by Stop.

Requires user role: USER

Parameters:
- CameraId(r): <1..7>
- Pan: <Left/Right/Stop>
- PanSpeed: <1..15>
- Tilt: <Down/Stop/Up>
- 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.
PresetId: Select preset 1 to 15.
CameraId: Give the camera id.

Requires user role: USER

Parameters:
- PresetId(r): <1..15>
- CameraId(r): <1..7>

Example:
- xCommand Camera PositionActivateFromPreset PresetId:1 CameraId:1
  OK
  *r PositionActivateFromPresetResult (status=OK):
  ** end
The Dial command

**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.

**Requires user role**: USER

**Parameters**:

- **Number(r)**: <S: 0, 255>
- **Protocol**: <H323/Sip>
- **CallRate**: <64..6000>
- **CallType**: <Audio/Video>

**Example**:

```plaintext
xCommand Dial Number:543210 Protocol:h323
OK
*r DialResult (status=OK):
CallId: 2
ConferenceId: 1
** end
```

The DTMFSend command

**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(r)**: <0..65534>
- **DTMFString(r)**: <S: 0, 32>

**Example**:

```plaintext
xCommand DTMFSend CallId:2 DTMFString:1234
*r DTMFSendResult (status=OK):
** 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(r): <0..65534>
- Value(r): <Left/Right/Up/Down/ZoomIn/ZoomOut>

**Example:**
```
xCommand FarEndControl Camera Move CallId:3 Value:left
* 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(r): <0..65534>

**Example:**
```
xCommand FarEndControl Camera Stop CallId:3
* 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(r): <0..65534>
- PresetId(r): <1..15>

**Example:**
```
xCommand FarEndControl Preset Activate CallId:3 PresetId:1
* 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(r): <0..65534>
- PresetId(r): <0..15>

**Example:**
```
xCommand FarEndControl Preset Store CallId:3 PresetId:1
* 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(r)**: <0..65534>
- **SourceId**: <0..15>

**Example**:

```
xCommand FarEndControl Source Select CallId: 3 SourceId: 1
```

```
xStatus Call
```

```
* r FECCSelectSourceResult (status=OK):
** end
```

---

**The GPIO command**

**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
```

```
xStatus GPIO ManualState
```

```
* 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 3 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..3>
- **ServerUrl**: <S: 1, 2048>
- **Expression**: <S: 1, 256>

**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
```

```
*r FeedbackRegisterResult (status=OK):
FeedbackSlot: 1
** end
```

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

**Requires user role**: ADMIN

**Parameters**:
- **FeedbackSlot**: <1..3>

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

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

**Example**:
```
xCommand Key Press Key:Home
```

```
*r KeyPressResult (status=OK):
** end
```

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

**Requires user role**: ADMIN

**Parameters**:
- **Key**: <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:** <S: 0, 255>
- **Duration:** <0..3600>

**Example:**

```plaintext
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:**

```plaintext
xCommand Message Alert Clear
OK
*r MessageAlertClearResult (status=OK):
** end
```

**xCommand Message TextLine Display**

Display a text line on screen. Optional 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:** <S: 0, 140>
- **X:** <1..10000>
- **Y:** <1..10000>
- **Duration:** <0..3600>

**Example:**

```plaintext
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:**

```plaintext
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:** <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
FeedbackId: <S: 0, 255>
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:** <0..5>

**Example:**

```
xCommand Message Prompt Response
OptionId: 1
FeedbackId: <S: 0, 255>
OK
*r MessagePromptResponseResult (status=OK):
  ** end
```
**xCommand Message Echo**
An event is generated when the command is executed. Use the xFeedback command to monitor
the feedback. Read more about the xFeedback command in the API introduction section in this
guide.

Text: Enter the text to be echoed.

Requires user role: **ADMIN**

Parameters:
- Text: <S: 0, 140>

Example:
```
xCommand Message Echo Text: "The text to be echoed." OK 
```
```
xFeedback Result (status=OK)
```
```
** end
```

**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: **USER**

Parameters:
- FolderId(r): <S: 0, 255>
- Name: <S: 0, 255>
- ParentFolderId: <S: 0, 255>

Example:
```
xCommand Phonebook Folder Add Name: "New York Office" OK
```
```
xPhonebookFolderAddResult (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: **USER**

Parameters:
- FolderId(r): <S: 0, 255>
- Name: <S: 0, 255>
- ParentFolderId: <S: 0, 255>

Example:
```
xCommand Phonebook Folder Modify FolderId:localGroupId-3 Name: "New York Head Office" OK
```
```
xPhonebookFolderModifyResult (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: **USER**

Parameters:

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

Example:

```plaintext
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.
Device: Select the device type.
ShowInBuddyList: Select if the contact should be listed in the buddy list.
SpeedDialNumber: Enter a speed dial number for the contact.

Requires user role: **USER**

Parameters:

Name(r): <S: 0, 255>
FolderId: <S: 0, 255>
ImageURL: <S: 0, 255>
Title: <S: 0, 255>
Number: <S: 0, 255>
Protocol: <H323/SIP>
CallRate: <0..65534>
Device: <Mobile/Other/Telephone/Video>
ShowInBuddyList: <False/True>
SpeedDialNumber: <S: 0, 255>

Example:

```plaintext
xCommand Phonebook Contact Add Name: "John Doe" Number:12345
ShowInBuddyList:True
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.
- **ShowInBuddyList:** Select if the contact should be listed in the buddy list.
- **SpeedDialNumber:** Enter a speed dial number for the contact.

**Requires user role:** USER

**Parameters:**
- **ContactId:** <S: 0, 255>
- **Name:** <S: 0, 255>
- **FolderId:** <S: 0, 255>
- **ImageURL:** <S: 0, 255>
- **Title:** <S: 0, 255>
- **ShowInBuddyList:** <False/True>
- **SpeedDialNumber:** <S: 0, 255>

**Example:**

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

```
*r PhonebookContactModifyResult (status=OK):
** end 
```

**xCommand Phonebook Contact Method 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:** The phone number or address of the contact.
- **Protocol:** Select H323 or SIP protocol.
- **CallRate:** Set a call rate.

**Requires user role:** USER

**Parameters:**
- **ContactId:** <S: 0, 255>
- **Device:** <Mobile/Other/Telephone/Video>
- **Number:** <S: 0, 255>
- **Protocol:** <H323/SIP>
- **CallRate:** <0..65534>

**Example:**

```
xCommand Phonebook ContactMethod Add ContactId:localContactId-2 Number:54321 Protocol:H323 OK 
```

```
*r PhonebookContactMethodAddResult (status=OK):
Name: 1 
** end 
```

**xCommand Phonebook Contact Method 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:** USER

**Parameters:**
- **ContactId:** <S: 0, 255>
- **ContactMethodId:** <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.

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

Offset: Get records starting with this offset in a search. Default 0. Used together with Limit to support paging.

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.

Requires user role: USER

Parameters:

PhonebookId: <S: 0, 255>
PhonebookType: <Corporate/Local>
SearchString: <S: 0, 255>
SearchField: <Name/Number>
FolderId: <S: 0, 255>
Offset: <0..65534>
Limit: <0..65534>

Example:

xCommand Phonebook Search PhonebookType:Corporate Limit:2
FolderId: "corporate_119"

OK

* ResultSet Contact 1 ContactMethod 2 Protocol: SIP
* ResultSet Contact 2 ContactId: "e__9468400"
* ResultSet Contact 2 ContactMethod 1 ContactMethodId: "3"
* ResultSet Contact 2 ContactMethod 1 Number: "H323:John.Doe.Office@company.com"
* ResultSet Contact 2 ContactMethod 1 Protocol: H323
* ResultSet Contact 2 ContactMethod 2 ContactMethodId: "4"
* ResultSet Contact 2 ContactMethod 2 Number: "SIP:John.Doe.Office@company.com"
* 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: \(<1..5>\)

Example:
```cisco
xCommand Presentation Start PresentationSource:2
OK
** end
```

**xCommand Presentation Stop**
Stop the media stream from the presentation source.

Requires user role: USER

Example:
```cisco
xCommand Presentation Stop
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:
```cisco
xCommand Preset Store PresetId:3 Type:Camera Description:"Left view"
OK
** end
```

**xCommand Preset Activate**
Activate one of the local presets.

Requires user role: USER

Parameters:
- PresetId(r): \(<1..15>\)

Example:
```cisco
xCommand Preset Activate PresetId:3
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. A reboot of the codec is required to complete the software upgrade.

AutoComplete On: The software upgrade is automatically completed, with a reboot of the codec.
AutoComplete Off: The software upgrade is not completed, and must be completed by running the xCommand Provisioning CompleteUpgrade command.

Requires user role: USER

Parameters:
  AutoComplete: <On/Off>

Example:
  xCommand Provisioning StartUpgrade AutoComplete:On
  *r StartUpgradeResult (status=OK):
  ** end

xCommand Provisioning CompleteUpgrade
Completes the software upgrade. This will reboot the codec. NOTE: This command is required if the xCommand Provisioning StartUpgrade has been run with the AutoComplete set to Off.

Requires user role: USER

Parameters:
  None

Example:
  xCommand Provisioning CompleteUpgrade
  *r CompleteUpgradeResult (status=OK):
  ** end

xCommand Provisioning CancelUpgrade
Cancel a software update in progress.

Requires user role: USER

Parameters:
  None

Example:
  xCommand Provisioning CancelUpgrade
  *r CancelUpgradeResult (status=OK):
  ** end
The SStringSend command

**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: 1, 2048>
- CallId: <0..65534>

Example:
```
xCommand SStringSend CallId:4 Message:"This is a test"
```
```
*r SStringSendResult (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, 255>

**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
```
```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, followed by an automatic reboot of the codec. 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.

Requires user role: ADMIN

Parameters:
Confirm(r): <Yes>

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
*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

**Parameters:**

- Name: <S: 0, 255>

**Example:**

```
xCommand SystemUnit ConfigurationProfile CancelChange Name: "My_ConfigurationProfile_1"
*ConfigurationProfileCancelChangeResult (status=OK):
** end
```

The TStringSend command

**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:** ADMIN

**Parameters:**

- CallId: <0..65534>
- Message: <S: 1, 1450>

**Example:**

```
xCommand TStringSend CallId:1 Message:"This is an example"
*T 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 ListLayoutFamily
Display a list of the layout families. If no LayoutFamilyId or DescriptorId is given, all layout families will be listed. Run the xCommand Video Layout ListLayoutFamily command once to see the LayoutFamilyIds or DescriptorIds.
LayoutFamilyId: A unique identifier of the layout family.
DescriptorId: A unique identifier of the descriptor.
Requires user role: USER
Parameters:
  LayoutFamilyId: <0..65534>
  DescriptorId: <0..65534>

Example:
xCommand Video Layout ListLayoutFamily
  //Comment: Displays the complete list of layout families.//
  ** 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 ListLayoutGraphic
Displays a list of all the defined layout graphics.
LayoutGraphicId: A unique identifier of the layout graphic. Run the xCommand Video Layout ListLayoutGraphic command once to see the LayoutGraphicIds.
FrameId: A unique identifier for the frame, which was created when the xCommand Video Layout Frame Add command was issued.
Requires user role: USER
Parameters:
  LayoutGraphicId: <0..65534>
  FrameId: <0..65534>

Example:
xCommand Video Layout ListLayoutGraphic
  //Comment: Displays the complete list of layout graphics.//
  ** 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
*xr 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
*xr 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
*xr 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.
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: <1..65535>
PositionX(r): <0..10000>
PositionY(r): <0..10000>
xCommand Video Layout Frame Add

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.

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
```

*x Command Video Layout Frame Remove

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.

Example:
```
xCommand Video Layout Frame Remove LayoutId:1 FrameId:1
```

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.

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.

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.

Requires user role: USER

Example:
```
xCommand Video Layout Frame Update LayoutId:1 PositionX:100 PositionY:100 Width:9800 Height:9800 Layer:1 Border:off VideoSourceType:localInput
VideoSourceId:1
OK
```

*x Command 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.

Example:
```
xCommand Video Layout Frame Remove LayoutId:1 FrameId:1
```

*x Command Video Layout Frame Update

Example:
Width(r): <1..10000>
Height(r): <1..10000>
Layer(r): <1..5>
Border: <On/Off>
VideoSourceId: <0..256>
VideoSourceType(r): <Graphic/LocalInput/LocalMain/LocalPresentation/MostSpeaking/OtherMain/OwnMain/Presentation/RemoteMain/RemotePresentation/VideoFile>

Example:
```
xCommand Video Layout Frame Update LayoutId:1 FrameId:1
VideoSourceType:localInput VideoSourceId:1
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
```

** 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.

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

** Example:
```
xCommand Video Layout AssignLocalOutput OutputId:1 LayoutId:2
OK
```

** 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.

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.

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>
- LayoutId(r): <1..2147483647>

** Example:
```
xCommand Video Layout AssignCall CallId:1 LayoutId:2
OK
```

** 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.

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

- **Requires user role:** USER

- **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: <0..65534>
  - OutputId: <0..65534>

- **Example:**
  ```
  xCommand Video Layout UnAssign CallId: 1 OutputId: 1
  OK
  *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.
- **LayoutId:** A unique identifier for the layout, which was created when the `xCommand Video Layout Add` command was issued.

- **Requires user role:** USER

- **Parameters:**
  - OutputId: <0..65534>
  - LayoutId: <1..2147483647>

- **Example:**
  ```
  xCommand Video Layout UnAssignLocalOutput OutputId:1 LayoutId:2
  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.
- **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: <0..65534>
  - LayoutId: <1..2147483647>

- **Example:**
  ```
  xCommand Video Layout UnAssignCall CallId:1 LayoutId:2
  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.

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 UnAssignPresentation LayoutId:2
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:** USER

**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.

CustomAutoMode: 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/CustomAutoMode/Default>

**Example:**
```
xCommand Video Layout LoadDb Type: Default
OK
* r VideoLayoutLoadDbResult (status=OK):
  ** end
```
The Experimental commands
The Experimental commands can be used 'as is' and will not be further documented.
NOTE: The Experimental commands are likely to change.

xCommand Experimental Audio LocalInput Mute
NOTE: This Experimental command can be used 'as is' and will not be further documented. The
Experimental settings WILL change.
Mute the local audio input.
Requires user role: ADMIN
Parameters:
  ConferenceId(r): <0..65534>
Example:
  xCommand Experimental Audio LocalInput Mute ConferenceID:1
  *r AudioLocalInputMuteResult (status=OK):
  ** end

The Experimental commands
The Experimental commands can be used 'as is' and will not be further documented.
The Experimental settings WILL change.
NOTE: The Experimental commands are likely to change.

xCommand Experimental Audio LocalInput Unmute
NOTE: This Experimental command can be used 'as is' and will not be further documented. The
Experimental settings WILL change.
Unmute the local audio input.
Requires user role: ADMIN
Parameters:
  ConferenceId(r): <0..65534>
Example:
  xCommand Experimental Audio LocalInput Unmute ConferenceID:1
  *r AudioLocalInputUnmuteResult (status=OK):
  ** end

xCommand Experimental Audio RemoteInput Mute
NOTE: This Experimental command can be used 'as is' and will not be further documented. The
Experimental settings WILL change.
Mute the remote audio input.
Requires user role: ADMIN
Parameters:
  ConferenceId(r): <0..65534>
Example:
  xCommand Experimental Audio RemoteInput Mute ConferenceID:1
  *r AudioRemoteInputMuteResult (status=OK):
  ** end

xCommand Experimental Audio RemoteInput Unmute
NOTE: This Experimental command can be used 'as is' and will not be further documented. The
Experimental settings WILL change.
Unmute the remote audio input.
Requires user role: ADMIN
Parameters:
  ConferenceId(r): <0..65534>
Example:
  xCommand Experimental Audio RemoteInput Unmute ConferenceID:1
  *r AudioRemoteInputUnmuteResult (status=OK):
  ** end

xCommand Experimental Audio RemoteOutput Mute
NOTE: This Experimental command can be used 'as is' and will not be further documented. The
Experimental settings WILL change.
Mute the remote audio output.
Requires user role: ADMIN
Parameters:
  ConferenceId(r): <0..65534>
Example:
  xCommand Experimental Audio RemoteOutput Mute ConferenceID:1
  *r AudioRemoteOutputMuteResult (status=OK):
  ** end

xCommand Experimental Audio RemoteOutput Unmute
NOTE: This Experimental command can be used 'as is' and will not be further documented. The
Experimental settings WILL change.
Unmute the remote audio output.
Requires user role: ADMIN
Parameters:
  ConferenceId(r): <0..65534>
Example:
  xCommand Experimental Audio RemoteOutput Unmute ConferenceID:1
  *r AudioRemoteOutputUnmuteResult (status=OK):
  ** end
**xCommand Experimental Audio LocalOutput Mute**

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Mute the remote local output.

**Requires user role:** ADMIN

**Parameters:**
- ConferenceId(r): <0..65534>

**Example:**
```
xCommand Experimental Audio LocalOutput Mute ConferenceID:1
```
```
*r AudioLocalOutputMuteResult (status=OK):
** end
```

**xCommand Experimental Audio LocalOutput Unmute**

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Unmute the remote local output.

**Requires user role:** ADMIN

**Parameters:**
- ConferenceId(r): <0..65534>

**Example:**
```
xCommand Experimental Audio LocalOutput Unmute ConferenceID:1
```
```
*r AudioLocalOutputUnmuteResult (status=OK):
** end
```

**xCommand Experimental Audio StereoEchoCancellation**

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Turn on/off the stereo echo cancellation.

**Requires user role:** ADMIN

**Parameters:**
- Mode(r): <On/Off>

**Example:**
```
xCommand Experimental Audio StereoEchoCancellation Mode: Off
```
```
*r AudioStereoEchoCancellationModeResult (status=OK):
** end
```

**xCommand Experimental VideoIndicator**

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

**Requires user role:** ADMIN

**Parameters:**
- VideoSourceId(r): <0..65534>
- Show(r): <On/Off>
- PositionX(r): <0..10000>
- PositionY(r): <0..10000>

**Example:**
```
xCommand Experimental VideoIndicator VideoSourceId:0 Show:Off PositionX:0 PositionY:0
```
```
*r VideoIndicatorResult (status=OK):
** end
```

**xCommand Experimental Camera DirectIRControl**

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

**Requires user role:** ADMIN

**Parameters:**
- CameraId(r): <1..7>
- OnOff(r): <On/Off>

**Example:**
```
xCommand Experimental Camera DirectIRControl CameraId:1 OnOff:Off
```
```
*r CameraDirectIRControlResult (status=OK):
** end
```

**xCommand Experimental SetLowLevel**

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

**Requires user role:** ADMIN

**Parameters:**
- Mode(r): <On/Off>

**Example:**
```
xCommand Experimental SetLowLevel Mode:Off
```
```
*r SetLowLevelResult (status=OK):
** end
```
### xCommand Experimental Call UnattendedTransfer

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

**Requires user role:** ADMIN

**Parameters:**
- CallId(r): <0..65534>
- Number(r): <S: 0, 255>

**Example:**
```
xCommand Experimental Call UnattendedTransfer CallId:321 Number:12345678
```
```
*r CallUnattendedTransferResult (status=OK):
** end
```

### xCommand Experimental Call AttendedTransfer

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

**Requires user role:** ADMIN

**Parameters:**
- FirstCallId(r): <0..65534>
- SecondCallId(r): <0..65534>

**Example:**
```
xCommand Experimental Call AttendedTransfer FirstCallId:321 SecondCallId:322
```
```
*r CallAttendedTransferResult (status=OK):
** end
```

### xCommand Experimental UserInput

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

**Requires user role:** ADMIN

**Parameters:**
- CallId(r): <0..65534>
- Device(r): <Mouse/NavKey>
- Button(r): <0..65534>
- Action: <Click/Click2/Click3/None/Press/Release>
- PositionX: <0..10000>
- PositionY: <0..10000>

**Example:**
```
xCommand Experimental UserInput CallId:0 Device:NavKey Button:0
```
```
*r UserInputResult (status=OK):
** end
```

### xCommand Experimental TakeWebSnapshot

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

**Requires user role:** ADMIN

**Example:**
```
xCommand Experimental TakeWebSnapshot
```
```
*r TakeWebSnapshotResult (status=OK):
** end
Chapter 5

Description of the xStatus commands
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 Input RemoteInput 8 AGC: "Off"
* 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
```
**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:**

<On/Off>

**Example:**

```plaintext
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:**

```plaintext
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.

**Value space of the result returned:**

<On/Off>

**Example:**

```plaintext
xStatus Audio Input LocalInput 1 Mute
*s Audio Input LocalInput 1 Mute: "Off"
** end
```

---

**xStatus Audio Microphones Mute**

Shows the microphones mute mode.

**Value space of the result returned:**

<On/Off>

**Example:**

```plaintext
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:**

```plaintext
xStatus Audio Volume
*s Audio Volume: 70
** end
```
xStatus Audio Input LocalInput [1..n] Channels

States if the local input channels are mixed 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 Input RemoteInput [1..n] AGC

Shows the AGC (Automatic Gain Control) mode on the remote 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:
<On/Off>

Example:
    xStatus Audio Input RemoteInput 8 AGC
    *s Audio Input RemoteInput 8 AGC: "Off"
    ** 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:
<S: 0, 255>

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 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:*
```c
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:*
```c
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 input 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:*
```c
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:*
```c
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:*
```c
xStatus Audio Output RemoteOutput 9 Input 1 Gain
*s Audio Output RemoteOutput 9 Input 1 Gain: 0
** end
```

**xStatus Audio Module [1..n] 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:*
```c
xStatus Audio Module 1 Type
*s Audio Module 1 Type: DigitalNAM
** 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: 768
*s Call 27 ReceiveCallRate: 4000
*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:
<Dialling/Connecting/Ringing/Connected/Idle>

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:**
```plaintext
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:**
```plaintext
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:**
<S: 0, 100>

**Example:**
```plaintext
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:**
<S: 0, 100>

**Example:**
```plaintext
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:**
<S: 0, 100>

**Example:**
```plaintext
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 value>

**Example:**
```plaintext
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 value>

**Example:**
```plaintext
xStatus Call 27 ReceiveCallRate
*s Call 27 ReceiveCallRate: 968
** 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:**
```plaintext
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 value>

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: "S0000000"
  *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"
xStatus Camera [1..7] Connected
Shows if the camera is connected or not.
Value space of the result returned:
<Ture/False>
Example:
xStatus Camera 1 Connected
*t Camera 1 Connected: True
** end

xStatus Camera [1..7] HardwareID
Shows the hardware identity of the camera.
Value space of the result returned:
<S: 0, 100>
Example:
xStatus Camera 1 HardwareID
*t Camera 1 HardwareID: "50000000"
** end

xStatus Camera [1..7] Manufacturer
Shows the manufacturer of the camera.
Value space of the result returned:
<S: 0, 100>
Example:
xStatus Camera 1 Manufacturer
*t Camera 1 Manufacturer: "TANDBERG"
** end

xStatus Camera [1..7] Model
Shows the camera model.
Value space of the result returned:
<S: 0, 100>
Example:
xStatus Camera 1 Model
*t 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:
<S: 0, 100>
Example:
xStatus Camera 1 SoftwareID
*t Camera 1 SoftwareID: "S01718-4.0FINAL [ID:40063] 2010-10-20"
** end

xStatus Camera [1..7] IpAddress
Shows the camera IP address.
Value space of the result returned:
<S: 0, 100>
Example:
xStatus Camera 1 IpAddress
*t 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:
<S: 0, 100>
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:
<S: 0, 100>
Example:
    xStatus Camera 1 Capabilities Options
    "s Camera 1 Capabilities Options: ""ptzf"
    ** 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:
```plaintext
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 Site 2 MicrophonesMuted: True
*s Conference Site 2 Capabilities Presentation: True
** end
```

**xStatus Conference Presentation Mode**
Shows the status of the secondary video stream.

Value space of the result returned:
<Off/Sending/Receiving>

Example:
```plaintext
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:
<S: 0, 10>

Example:
```plaintext
xStatus Conference Presentation Protocol
*s Conference Presentation Protocol: "H264"
** 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] 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
```

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
```

xStatus Diagnostics Call [1..n] Channels IncomingAudioChannel [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 value>

Example:

xStatus Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 Jitter
*s Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 Jitter: 0
** end

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

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 value>

Example:

xStatus Diagnostics Call 27 Channels OutgoingAudioChannel 336 Netstat 1 Packets
*s Diagnostics Call 27 Channels OutgoingAudioChannel 336 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 packets lost in the incoming/outgoing channels.

Value space of the result returned:
<Integer value>

Example:
```bash
xStatus Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 Loss
```
```
27 Diagnostics Call 27 Channels IncomingAudioChannel 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

Shows the packets lost during the last interval for the incoming channels.

Value space of the result returned:
<Integer value>

Example:
```bash
xStatus Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 LastIntervalLost
```
```
27 Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 LastIntervalLost: 96
* 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

Shows the packets received during the last interval for the incoming channels.

Value space of the result returned:
<Integer value>

Example:
```bash
xStatus Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 LastIntervalReceived
```
```
27 Diagnostics Call 27 Channels IncomingAudioChannel 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 packets dropped in the incoming/outgoing channel.

Value space of the result returned:
<Integer value>

Example:
```bash
xStatus Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 Drop
```
```
27 Diagnostics Call 27 Channels IncomingAudioChannel 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

Showed the number of bytes received/sent in the incoming/outgoing channel.

Value space of the result returned:

<Integer value>

Example:

xStatus Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 Bytes

*s Diagnostics Call 27 Channels IncomingAudioChannel 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 value>

Example:

xStatus Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 ChannelRate

*s Diagnostics Call 27 Channels IncomingAudioChannel 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 value>

Example:

xStatus Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 MaxJitter

xStatus GPIO Pin 1 State

* s GPIO Pin 1 State: High
** 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:
<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:
<S: 0, 100>

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 the gatekeeper.

Value space of the result returned:
<Integer value>

Example:
xStatus H323 Gatekeeper Port
*s H323 Gatekeeper Port: 1719
** 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: ""
  *s HttpFeedback 1 Expression: ""
  *s HttpFeedback 1 Expression: ""
  *s HttpFeedback 1 Expression: ""
  ** end
```

xStatus HttpFeedback [1..3] 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:
<S: 0, 100>

Example:
xStatus HttpFeedback 1 URL
```
xStatus HttpFeedback 1 URL
  ** end
```

xStatus HttpFeedback [1..3] Expression
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:
<S: 0..256>

Example:
xStatus HttpFeedback 3 Expression
```
xStatus HttpFeedback 3 Expression
  *s HttpFeedback 3 Expression: "/History/CallLog/History"
  *s HttpFeedback 3 Expression: "/Status/Call[Status='connected']"
  *s HttpFeedback 3 Expression: "/Status/H323/Gatekeeper"
  *s HttpFeedback 3 Expression: "/Status/Ethernet"
  *s HttpFeedback 3 Expression: "/Event/CallSuccessful"
  *s HttpFeedback 3 Expression: ""
  *s HttpFeedback 3 Expression: ""
  *s HttpFeedback 3 Expression: ""
  *s HttpFeedback 3 Expression: ""
  *s HttpFeedback 3 Expression: ""
  *s HttpFeedback 3 Expression: ""
  *s HttpFeedback 3 Expression: ""
  *s HttpFeedback 3 Expression: ""
  *s HttpFeedback 3 Expression: ""
  ** end
```
The MedialChannels 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 27 IncomingAudioChannel 327 Encryption Status: Off
  *s MediaChannels Call 27 IncomingAudioChannel 327 Audio Protocol: AACLD
  *s MediaChannels Call 27 IncomingAudioChannel 327 Audio Channels: 1
  *s MediaChannels Call 27 IncomingVideoChannel 330 Encryption Status: Off
  *s MediaChannels Call 27 IncomingVideoChannel 330 ChannelRole: Main
  *s MediaChannels Call 27 IncomingVideoChannel 330 Video Protocol: H264
  *s MediaChannels Call 27 IncomingVideoChannel 330 Video FrameRate: 25
  *s MediaChannels Call 27 IncomingVideoChannel 330 Video ResolutionX: 352
  *s MediaChannels Call 27 IncomingVideoChannel 330 Video ResolutionY: 288
  *s MediaChannels Call 27 IncomingVideoChannel 333 Encryption Status: Off
  *s MediaChannels Call 27 IncomingVideoChannel 333 ChannelRole: Presentation
  *s MediaChannels Call 27 IncomingVideoChannel 333 Video Protocol: Off
  *s MediaChannels Call 27 IncomingVideoChannel 333 Video FrameRate: 0
  *s MediaChannels Call 27 IncomingVideoChannel 333 Video ResolutionX: 0
  *s MediaChannels Call 27 IncomingVideoChannel 333 Video ResolutionY: 0
  *s MediaChannels Call 27 OutgoingAudioChannel 328 Encryption Status: Off
  *s MediaChannels Call 27 OutgoingAudioChannel 328 Audio Protocol: AACLD
  *s MediaChannels Call 27 OutgoingAudioChannel 328 Audio Channels: 1
  *s MediaChannels Call 27 OutgoingVideoChannel 331 Encryption Status: Off
  *s MediaChannels Call 27 OutgoingVideoChannel 331 ChannelRole: Main
  *s MediaChannels Call 27 OutgoingVideoChannel 331 Video Protocol: H264
  *s MediaChannels Call 27 OutgoingVideoChannel 331 Video FrameRate: 30
  *s MediaChannels Call 27 OutgoingVideoChannel 331 Video ResolutionX: 1024
  *s MediaChannels Call 27 OutgoingVideoChannel 331 Video ResolutionY: 576
  *s MediaChannels Call 27 OutgoingVideoChannel 334 Encryption Status: Off
  *s MediaChannels Call 27 OutgoingVideoChannel 334 ChannelRole: Presentation
  *s MediaChannels Call 27 OutgoingVideoChannel 334 Video Protocol: Off
  *s MediaChannels Call 27 OutgoingVideoChannel 334 Video FrameRate: 30
  *s MediaChannels Call 27 OutgoingVideoChannel 334 Video ResolutionX: 0
  *s MediaChannels Call 27 OutgoingVideoChannel 334 Video ResolutionY: 0
```

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
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Audio Protocol
Shows the audio algorithm for the incoming audio channel.

**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
```

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
```

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
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] ChannelRole
Channel role of the incoming video channel.

**Example:**
```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 ChannelRole: Main
  *s MediaChannels Call 27 IncomingVideoChannel 330 ChannelRole: Main
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Video Protocol
Video compression format of the incoming video channel.

**Example:**
```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Video Protocol: H264
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Video FrameRate
Video frame rate of the incoming video channel.

**Example:**
```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Video FrameRate: 25
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Video ResolutionX
Video width of the incoming video channel.

**Example:**
```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Video ResolutionX: 352
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Video ResolutionY
Video height of the incoming video channel.

**Example:**
```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Video ResolutionY: 288
```

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
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Audio Protocol
Shows the audio algorithm for the outgoing audio channel.

**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
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Audio Channels
Audio channels of the outgoing audio channel.

**Example:**
```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Audio Channels: 1
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] ChannelRole
Channel role of the outgoing video channel.

**Example:**
```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 ChannelRole: Main
  *s MediaChannels Call 27 OutgoingVideoChannel 331 ChannelRole: Main
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Video Protocol
Video compression format of the outgoing video channel.

**Example:**
```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video Protocol: H264
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Video FrameRate
Video frame rate of the outgoing video channel.

**Example:**
```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video FrameRate: 30
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Video ResolutionX
Video width of the outgoing video channel.

**Example:**
```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video ResolutionX: 1024
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Video ResolutionY
Video height of the outgoing video channel.

**Example:**
```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video ResolutionY: 576
```

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
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Video Mute
Video mute status of the outgoing video channel.

**Value space of the result returned:**

<True/False>

**Example:**
```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video Mute
  *s MediaChannels Call 27 OutgoingVideoChannel 331 Video Mute: True
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Video Channels
Video channels of the outgoing video channel.

**Example:**
```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video Channels: 1
```

**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 value>

Example:
```c
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Audio Channels
```
```c
*x MediaChannels Call 27 IncomingAudioChannel 327 Audio Channels: 1
** 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:
```c
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Encryption Status
```
```c
*x 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:
```c
xStatus MediaChannels Call 27 IncomingVideoChannel 330 ChannelRole
```
```c
*x 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.
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:
```c
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Video Protocol
```
```c
*x 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 value>

Example:
```c
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Video FrameRate
```
```c
*x 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 value>

Example:
```c
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Video ResolutionX
```
```c
*x 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 value>

Example:
```c
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Video ResolutionY
```
```c
*x MediaChannels Call 27 IncomingVideoChannel 330 Video ResolutionY: 448
** 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:
```c
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Encryption Status
```
```c
*x MediaChannels Call 27 OutgoingAudioChannel 328 Encryption Status: Off
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Video FrameRate
Shows the video frame rate of the outgoing channel.

Value space of the result returned:
<Integer value>

Example:
```c
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Video FrameRate
```
```c
*x MediaChannels Call 27 OutgoingAudioChannel 328 Video FrameRate: 25
** end
```
**xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Audio Protocol**

Shows the audio algorithm for the outgoing audio channel.

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.

G722.1: 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/G722.1/G711Mu/G711A>

**Example:**

```plaintext
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Audio Protocol
* 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 value>

**Example:**

```plaintext
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Audio Channels
* MediaChannels Call 27 OutgoingAudioChannel 328 Audio Channels: 1
** 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:**

```plaintext
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Encryption Status
* 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:**

```plaintext
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 ChannelRole
* 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.


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:**

```plaintext
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video Protocol
* 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 value>

**Example:**

```plaintext
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video FrameRate
* 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 value>

**Example:**

```plaintext
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video ResolutionX
* 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 value.>

Example:
\[
\text{xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video ResolutionY}
\]
\[
\text{'s MediaChannels Call 27 OutgoingVideoChannel 331 Video ResolutionY: 448}
\]

The Network status

xStatus Network

Shows the top level overview of the network status.

Example:
\[
\text{xStatus Network}
\]
\[
\text{'s Network 1 Ethernet MacAddress: "00:50:60:02:E7:D3"
's Network 1 Ethernet Speed: "1000full"
's Network 1 IPv4 Address: "192.0.2.149"
's Network 1 IPv4 SubnetMask: "255.255.255.0"
's Network 1 IPv4 Gateway: "192.0.2.10"
's Network 1 IPv4 DNS Domain Name: "www.example.com www.example.int"
's Network 1 IPv4 DNS Server 1 Address: "192.0.2.60"
's Network 1 IPv4 DNS Server 2 Address: "192.0.2.61"
's Network 1 IPv4 DNS Server 3 Address: ""
's Network 1 IPv4 DNS Server 4 Address: ""
's Network 1 IPv4 DNS Server 5 Address: ""
's Network 1 IPv6 Address: ""
's Network 1 IPv6 Gateway: ""
's Network 1 IPv4 MTU: 1500
\]

xStatus Network 1 Ethernet MacAddress

Shows the MAC (Media Access Control) address for the ethernet interface.

Value space of the result returned:
<S: 0, 100>

Example:
\[
\text{xStatus Network 1 Ethernet MacAddress}
\]
\[
\text{'s Network 1 Ethernet MacAddress: "00:50:60:02:FD:C7"
\]

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:
\[
\text{xStatus Network 1 Ethernet Speed}
\]
\[
\text{'s Network 1 Ethernet Speed: "100full"
\]
xStatus Network 1 IPv4 Address
Shows the IP address that uniquely identifies this system.
Value space of the result returned:
<S: 0, 100>
Example:
```plaintext
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 what subnet an IP address belongs to.
Value space of the result returned:
<S: 0, 100>
Example:
```plaintext
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 gateway.
Value space of the result returned:
<S: 0, 100>
Example:
```plaintext
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:
<S: 0, 100>
Example:
```plaintext
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:
<S: 0, 100>
Example:
```plaintext
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:
<S: 0, 100>
Example:
```plaintext
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:
<S: 0, 100>
Example:
```plaintext
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 value>
Example:
```plaintext
xStatus Network 1 MTU
*s Network 1 MTU: 1500
** 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
Specifies if a camera preset is stored at this position.

Value space of the result returned:
<True/False>

Example:
```
xStatus Preset 3 Defined
  *s Preset 3 Defined: True
** end
```

xStatus Preset [1..15] Type
Specifies the camera preset type.

Value space of the result returned:
<All/...>

Example:
```
xStatus Preset 3 Type
  *s Preset 3 Type: All
** end
```
The Provisioning status

**xStatus Provisioning**
Shows the top level overview of the provisioning status.

Example:
```plaintext
xStatus Provisioning
  *s Provisioning Status: Provisioned
  *s Provisioning Reason: ""
  *s Provisioning PendingUpgrade: False
** 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:
```plaintext
xStatus Provisioning Status
  *s Provisioning Status: Provisioned
** end
```

**xStatus Provisioning Reason**
Shows the cause when provisioning has failed.

Value space of the result returned:
<S: 0..80>

Example:
```plaintext
xStatus Provisioning Reason
  *s Provisioning Reason: ""
** 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: "firstname.lastname@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 Proxy 2 Status: Off
*s SIP Profile 1 Proxy 2 Address: ""
*s SIP Profile 1 Proxy 3 Status: Off
*s SIP Profile 1 Proxy 3 Address: ""
*s SIP Profile 1 Proxy 4 Status: Off
*s SIP Profile 1 Proxy 4 Address: "0"
*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: "firstname.lastname@company.com"
```

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:

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

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:**
<Ture/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:**
<S: 0, 100>

**Example:**
```
xStatus SIP Registration 1 Reason
  *S SIP Registration 1 Reason: "404 Not Found"
  ** end
```

**xStatus SIP Profile 1 Proxy [1..4] 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 Registration [1..n] URI**
Shows the URI used for registration to the SIP Proxy server.

**Value space of the result returned:**
<S: 0, 100>

**Example:**
```
xStatus SIP Registration 1 URI
  *S SIP Registration 1 URI: "firstname.lastname@company.com"
  ** end
```

**xStatus SIP Registration [1..n] Authentication**
Shows which authentication mechanism is used when registering to the SIP Proxy Server.

Digest: Uses the Digest access authentication method, as specified by RFC 2069.
NTLM: Uses the NTLM authentication method, which is a Microsoft authentication protocol.
Off: No authentication mechanism is used.

**Value space of the result returned:**
<Digest/NTLM/Off>

**Example:**
```
xStatus SIP Registration 1 Authentication
  *S SIP Registration 1 Authentication: Off
  ** end
```

**xStatus SIP Profile 1 Proxy [1..4] Reason**
Shows the reason why the SIP Profile 1 Proxy status has changed.

**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..4] Address
Shows the address of the SIP Proxy that the system communicates with.
Value space of the result returned:
<S: 0, 255>
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 authentication method, which is a Microsoft authentication protocol.
Off: No authentication mechanism is used.
Value space of the result returned:
<Digest/NTLM/Off>
Example:
  xStatus SIP Profile 1 Authentication  
  *s SIP Profile 1 Authentication: Off
  ** end

xStatus SIP Profile 1 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:
<S: 0, 100>
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:
<S: 0, 100>
Example:
  xStatus SIP Profile 1 Registration 1 URI  
  *s SIP Profile 1 Registration 1 URI: "firstname.lastname@company.com"
  ** end
The Sleep status

**xStatus Sleep Active**

Shows if the system is in sleep mode or not.

**Value space of the result returned:**

<On/Off>

**Example:**

```plaintext
xStatus Sleep Active
*s Sleep Active: Off
** 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:**

```plaintext
xStatus Standby Active
*s Standby Active: Off
** end
```
The SystemUnit status

**xStatus SystemUnit**

Shows the top level overview of the system unit status.

**Example:**

```plaintext
xStatus SystemUnit
*s SystemUnit ProductType: "Cisco TelePresence Codec"
*s SystemUnit ProductId: "Cisco TelePresence Codec C90"
*s SystemUnit ProductPlatform: "C90"
*s SystemUnit Uptime: 597095
*s SystemUnit Software Application: "Endpoint"
*s SystemUnit Software Version: "TC4.0"
*s SystemUnit Software Name: "s52000"
*s SystemUnit Software ReleaseDate: "2010-11-01"
*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 Hardware Module SerialNumber: "B1AD25A00003"
*s SystemUnit Hardware Module Identifier: "0"
*s SystemUnit Hardware MainBoard SerialNumber: "PH0497201"
*s SystemUnit Hardware MainBoard Identifier: "101401-3 [04]"
*s SystemUnit Hardware VideoBoard SerialNumber: "PH0497874"
*s SystemUnit Hardware VideoBoard Identifier: "101560-1 [02]"
*s SystemUnit Hardware AudioBoard SerialNumber: "N/A"
*s SystemUnit Hardware AudioBoard Identifier: "" 
*s SystemUnit Hardware BootSoftware: "U-Boot 2009.03-65"
*s SystemUnit State System: Initialized
*s SystemUnit State MaxNumberOfCalls: 3
*s SystemUnit State NumberOfActiveCalls: 3
*s SystemUnit State NumberOfSuspendedCalls: 0
*s SystemUnit State NumberOfInProgressCalls: 0
*s SystemUnit State Subsystem Application: Initialized
*s SystemUnit ContactInfo: "helpdesk@company.com"
** end
```

**xStatus SystemUnit ProductType**

Shows the product type.

**Value space of the result returned:**

<\S: 0, 100>

**Example:**

```plaintext
xStatus SystemUnit ProductType
*s SystemUnit ProductType: "Cisco TelePresence Codec"
** end
```

**xStatus SystemUnit ProductId**

Shows the product identity.

**Value space of the result returned:**

<\S: 0, 100>

**Example:**

```plaintext
xStatus SystemUnit ProductId
*s SystemUnit ProductId: "Cisco TelePresence Codec C90"
** end
```

**xStatus SystemUnit ProductPlatform**

Shows the product platform.

**Value space of the result returned:**

<\S: 0, 100>

**Example:**

```plaintext
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 value>

**Example:**

```plaintext
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:**
\[S: 0, 100]\n
**Example:**
```c
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:**
\[S: 0, 100]\n
**Example:**
```c
xStatus SystemUnit Software Version
*s SystemUnit Software Version: "TC3.1.0"
** end
```

### xStatus SystemUnit Software Name
Shows the name of the software that is installed on the codec.

**Value space of the result returned:**
\[S: 0, 100]\n
**Example:**
```c
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:**
\[S: 0, 100]\n
**Example:**
```c
xStatus SystemUnit Software ReleaseDate
*s SystemUnit Software ReleaseDate: "2010-04-30"
** end
```

### xStatus SystemUnit Software MaxVideoCalls
Shows the maximum number of simultaneous video calls that is supported.

**Value space of the result returned:**
\[Integer value]\n
**Example:**
```c
xStatus SystemUnit Software MaxVideoCalls
*s SystemUnit Software MaxVideoCalls: 3
** end
```

### xStatus SystemUnit Software MaxAudioCalls
Shows the maximum number of simultaneous audio calls that is supported.

**Value space of the result returned:**
\[Integer value]\n
**Example:**
```c
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:**
\[S: 0, 100]\n
**Example:**
```c
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:**
\[S: 0, 100]\n
**Example:**
```c
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:
<S: 0, 100>
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:
<S: 0, 100>
Example:
  xStatus SystemUnit Software OptionKeys PremiumResolution
  "s SystemUnit Software OptionKeys PremiumResolution: "true"
  ** end

xStatus SystemUnit Hardware Module SerialNumber
Shows the serial number of the hardware module in the codec.
Value space of the result returned:
<S: 0, 100>
Example:
  xStatus SystemUnit Hardware Module SerialNumber
  "s SystemUnit Hardware Module SerialNumber: "B1AD25A00002"
  ** end

xStatus SystemUnit Hardware Module Identifier
Shows the revision of the hardware module in the codec.
Value space of the result returned:
<S: 0, 100>
Example:
  xStatus SystemUnit Hardware Module Identifier
  "s SystemUnit Hardware Module Identifier: "1"
  ** end

xStatus SystemUnit Hardware MainBoard SerialNumber
Shows the serial number of the main board in the codec.
Value space of the result returned:
<S: 0, 100>
Example:
  xStatus SystemUnit Hardware MainBoard SerialNumber
  "s SystemUnit Hardware MainBoard SerialNumber: "PH0528833"
  ** end

xStatus SystemUnit Hardware MainBoard Identifier
Shows the revision of the main board in the codec.
Value space of the result returned:
<S: 0, 100>
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:
<S: 0, 100>
Example:
  xStatus SystemUnit Hardware VideoBoard SerialNumber
  "s SystemUnit Hardware VideoBoard SerialNumber: "PH0534914"
  ** end

xStatus SystemUnit Hardware VideoBoard Identifier
Shows the revision of the video board in the codec.
Value space of the result returned:
<S: 0, 100>
Example:
  xStatus SystemUnit Hardware VideoBoard Identifier
  "s SystemUnit Hardware VideoBoard Identifier: "103140-4 [07]"
  ** end
xStatus SystemUnit Hardware AudioBoard SerialNumber
Shows the serial number of the audio board in the codec.
Value space of the result returned:
<S: 0, 100>
Example:
```csharp
xStatus SystemUnit Hardware AudioBoard SerialNumber
*s SystemUnit Hardware AudioBoard SerialNumber: "TBD"
** end
```

xStatus SystemUnit Hardware AudioBoard Identifier
Shows the revision of the audio board in the codec.
Value space of the result returned:
<S: 0, 100>
Example:
```csharp
xStatus SystemUnit Hardware AudioBoard Identifier
*s SystemUnit Hardware AudioBoard Identifier: "101420-2 [No obj1.]
** end
```

xStatus SystemUnit Hardware BootSoftware
Shows the version of the boot software that is installed on the codec.
Value space of the result returned:
<S: 0, 100>
Example:
```csharp
xStatus SystemUnit Hardware BootSoftware
*s SystemUnit Hardware BootSoftware: "U-Boot 2010.04-30"
** 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:
```csharp
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:
```csharp
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:
```csharp
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:  
<S: 0, 100>
Example:
```
xStatus SystemUnit ContactInfo
*s SystemUnit ContactInfo: "firstname.lastname@company.com"
** end
```
The Time status

xStatus Time Zone Olson
Shows the current time zone on Olson format.

Value space of the result returned:
<S: 0, 100>

Example:

xStatus Time Zone Olson
*s Time Zone Olson: Europe/Berlin
** end

The Video Input status

xStatus Video Input
Shows the top level overview of the video input status.

Example:

xStatus Video Input
*s Video Input Source 1 Resolution Height: 1080
*s Video Input Source 1 Resolution Width: 1920
*s Video Input Source 1 Resolution RefreshRate: 60
*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 3 Resolution Height: 0
*s Video Input Source 3 Resolution Width: 0
*s Video Input Source 3 Resolution RefreshRate: 0
*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 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 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:
<Ture/False>

Example:
```
xStatus Video Input HDMI 1 Connected
*s Video Input HDMI 1 Connected: False
** 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 2 SignalState
*s Video Input HDMI 2 SignalState: OK
** end
```

xStatus Video Input HDMI [2..3] Connected
Shows if there is something connected to the HDMI connector. Not all connections can be detected.

Value space of the result returned:
<Ture/False>

Example:
```
xStatus Video Input HDMI 2 Connected
*s Video Input HDMI 2 Connected: False
** 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:
<Ture/False>

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 DVI [2..3] SignalState
The Video Output status

**xStatus Video Output**
Shows the top level overview of the video output status.

**Example:**
```
xStatus Video Output
*s Video Output HDMI 1 Resolution Height: 1080
*s Video Output HDMI 1 Resolution Width: 1920
*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 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 Input DVI [2..3] SignalState**
Shows the signal state of the DVI input connector.

**Value space of the result returned:**
< ??? >

**Example:**
```
xStatus Video Input DVI 2 SignalState
*s Video Input DVI 2 SignalState: ???
** 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>

**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 of the Legacy, which are the Y/Comp and C connectors.

**Value space of the result returned:**
< ??? >

**Example:**
```
xStatus Video Input Legacy 3 SignalState
*s Video Input Legacy 3 SignalState: ???
** end
```
xStatus Video Output HDMI [1] Resolution RefreshRate
Shows the resolution width (in pixels) 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
The Video Layout status

xStatus Video Layout
Shows the top level overview of the video status.

Example:
```plaintext
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 FamilyNumber: 1027
*s Video Layout Site 1 Output 1 GraphicName: "1top-1small"
*s Video Layout Site 1 Output 1 GraphicNumber: 1017
*s Video Layout Site 1 Output 1 Descriptor: 4
*s Video Layout Site 1 Output 1 DescriptorOutput: 1
*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..3] and Frames [1..6].
** end

xStatus Video Layout PresentationView
Returns information about the presentation view mode.

Value space of the result returned:
<Default/Maximized/Minimized>

Example:
```plaintext
xStatus Video Layout PresentationView
*s Video Layout PresentationView: "full"
** end
```
**xStatus Video Layout Site [1..n] Output [1..3] GraphicNumber**
Shows the number of the graphic layout. The number identifies the layout used right now at the specified output.

Value space of the result returned:
<Integer value>

Example:
```plaintext
xStatus Video Layout Site 1 Output 1 GraphicNumber
's Video Layout Site 1 Output 1 GraphicNumber: 1037
** end
```

**xStatus Video Layout Site [1..n] Output [1..3] Descriptor**
Shows the descriptor of the layout.

Value space of the result returned:
<Integer value>

Example:
```plaintext
xStatus Video Layout Site 1 Output 1 Descriptor
's Video Layout Site 1 Output 1 Descriptor: 3
** end
```

**xStatus Video Layout Site [1..n] Output [1..3] DescriptorOutput**
Shows the descriptor output of the layout.

Value space of the result returned:
<Integer value>

Example:
```plaintext
xStatus Video Layout Site 1 Output 1 DescriptorOutput
's Video Layout Site 1 Output 1 DescriptorOutput: 1
** 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:
```plaintext
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:
```plaintext
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:
```plaintext
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:
```plaintext
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..3>

Example:
```plaintext
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:
<S:0,100>

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
```


Value space of the result returned:
<S: 0, 200>

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 commands are beta preview features and can be used 'as is'. They are not fully documented.

NOTE: The Experimental commands are likely to change without further notice.

xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] PacketRecovery ProtectionPackets MediaPacketsCovered

xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] PacketRecovery ProtectionPackets MediaPacketsCovered

xStatus Experimental Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] PacketRecovery ProtectionPackets MediaPacketsCovered

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Value space of the result returned:
<Integer value>

Example:
```
xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327 PacketRecovery ProtectionPackets MediaPacketsCovered
*s Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327 PacketRecovery ProtectionPackets MediaPacketsCovered: 0
** end
```

xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] PacketRecovery ProtectionPackets Received

xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] PacketRecovery ProtectionPackets Received

xStatus Experimental Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] PacketRecovery ProtectionPackets Received

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Value space of the result returned:
<Integer value>

Example:
```
xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327 PacketRecovery ProtectionPackets Received
*s Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327 PacketRecovery ProtectionPackets Received: 0
** end
```
PacketRecovery ProtectionPackets Received: 0
** end

xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n]
PacketRecovery ProtectionPackets Lost

xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n]
PacketRecovery ProtectionPackets Lost

xStatus Experimental Diagnostics Call [1..n] Channels IncomingDataChannel [1..n]
PacketRecovery ProtectionPackets Lost

Value space of the result returned:
<Integer value>

Example:

xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
PacketRecovery ProtectionPackets Lost
*s Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
PacketRecovery ProtectionPackets Lost: 0
** end

PacketRecovery MediaPackets Received

xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
PacketRecovery MediaPackets Received

* Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
PacketRecovery MediaPackets Received: 0
** end

PacketRecovery MediaPackets Lost

xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
PacketRecovery MediaPackets Lost

* Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
PacketRecovery MediaPackets Lost: 0
** end

PacketRecovery MediaPackets OutOfOrder

xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
PacketRecovery MediaPackets OutOfOrder

* Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
PacketRecovery MediaPackets OutOfOrder: 0
** end

xStatus Experimental Diagnostics Call 27 Channels IncomingVideoChannel 327
PacketRecovery MediaPackets OutOfOrder

xStatus Experimental Diagnostics Call 27 Channels IncomingDataChannel 327
PacketRecovery MediaPackets OutOfOrder

Value space of the result returned:
<Integer value>

Example:

xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
PacketRecovery MediaPackets OutOfOrder
*s Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
PacketRecovery MediaPackets OutOfOrder: 0
** end
xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n]
PacketRecovery MediaPackets OutOfOrder

Value space of the result returned:
<Integer value>

Example:
  xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets OutOfOrder
  *x Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets OutOfOrder: 0
  ** end

xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n]
PacketRecovery MediaPackets OutOfOrder

Value space of the result returned:
<Integer value>

Example:
  xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets OutOfOrder
  *x Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets OutOfOrder: 0
  ** end

xStatus Experimental Diagnostics Call [1..n] Channels IncomingDataChannel [1..n]
PacketRecovery MediaPackets OutOfOrder

Value space of the result returned:
<Integer value>

Example:
  xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets OutOfOrder
  *x Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets OutOfOrder: 0
  ** end

xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n]
PacketRecovery MediaPackets Recovered

Value space of the result returned:
<Integer value>

Example:
  xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets Recovered
  *x Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets Recovered: 0
  ** end

xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n]
PacketRecovery MediaPackets Recovered

Value space of the result returned:
<Integer value>

Example:
  xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets Recovered
  *x Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets Recovered: 0
  ** end

xStatus Experimental Diagnostics Call [1..n] Channels IncomingDataChannel [1..n]
PacketRecovery MediaPackets Recovered

Value space of the result returned:
<Integer value>

Example:
  xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets Recovered
  *x Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets Recovered: 0
  ** end

xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n]
PacketRecovery MediaPackets ConsecutiveLost

Value space of the result returned:
<Integer value>

Example:
  xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets ConsecutiveLost
  *x Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets ConsecutiveLost: 0
  ** end

xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n]
PacketRecovery MediaPackets ConsecutiveLost

Value space of the result returned:
<Integer value>

Example:
  xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets ConsecutiveLost
  *x Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets ConsecutiveLost: 0
  ** end

xStatus Experimental Diagnostics Call [1..n] Channels IncomingDataChannel [1..n]
PacketRecovery MediaPackets ConsecutiveLost

Value space of the result returned:
<Integer value>

Example:
  xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets ConsecutiveLost
  *x Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets ConsecutiveLost: 0
  ** end

xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n]
PacketRecovery MediaPackets Statistics ZeroLost

Value space of the result returned:
<Integer value>

Example:
  xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets Statistics ZeroLost
  *x Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets Statistics ZeroLost: 0
  ** end

xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n]
PacketRecovery MediaPackets Statistics ZeroLost

Value space of the result returned:
<Integer value>

Example:
  xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets Statistics ZeroLost
  *x Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets Statistics ZeroLost: 0
  ** end

xStatus Experimental Diagnostics Call [1..n] Channels IncomingDataChannel [1..n]
PacketRecovery MediaPackets Statistics ZeroLost

Value space of the result returned:
<Integer value>

Example:
  xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets Statistics ZeroLost
  *x Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327
  PacketRecovery MediaPackets Statistics ZeroLost: 0
  ** end
xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] PacketRecovery MediaPackets Statistics OneLost

xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] PacketRecovery MediaPackets Statistics OneLost

xStatus Experimental Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] PacketRecovery MediaPackets Statistics OneLost

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Value space of the result returned:
<Integer value>

Example:

```c
xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327 PacketRecovery MediaPackets Statistics OneLost
* Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327 PacketRecovery MediaPackets Statistics OneLost: 0
** end
```

xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] PacketRecovery MediaPackets Statistics TwoLost

xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] PacketRecovery MediaPackets Statistics TwoLost

xStatus Experimental Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] PacketRecovery MediaPackets Statistics TwoLost

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Value space of the result returned:
<Integer value>

Example:

```c
xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327 PacketRecovery MediaPackets Statistics TwoLost
* Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327 PacketRecovery MediaPackets Statistics TwoLost: 0
** end
```

xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] PacketRecovery MediaPackets Statistics ThreeLost

xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] PacketRecovery MediaPackets Statistics ThreeLost

xStatus Experimental Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] PacketRecovery MediaPackets Statistics ThreeLost

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Value space of the result returned:
<Integer value>

Example:

```c
xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327 PacketRecovery MediaPackets Statistics ThreeLost
* Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327 PacketRecovery MediaPackets Statistics ThreeLost: 0
** end
```

xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] PacketRecovery MediaPackets Statistics FourLost

xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] PacketRecovery MediaPackets Statistics FourLost

xStatus Experimental Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] PacketRecovery MediaPackets Statistics FourLost

NOTE: This Experimental command can be used 'as is' and will not be further documented. The Experimental settings WILL change.

Value space of the result returned:
<Integer value>

Example:

```c
xStatus Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327 PacketRecovery MediaPackets Statistics FourLost
* Experimental Diagnostics Call 27 Channels IncomingAudioChannel 327 PacketRecovery MediaPackets Statistics FourLost: 0
** end
```
xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n]
PacketRecovery MediaPackets Statistics MoreThanFourLost

xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n]
PacketRecovery MediaPackets Statistics MoreThanFourLost

NOTE: This Experimental command can be used 'as is' and will not be further documented. The
Experimental settings WILL change.

xStatus Experimental Diagnostics Call [1..n] Channels IncomingDataChannel [1..n]
PacketRecovery MediaPackets Statistics MoreThanFourLost

NOTE: This Experimental command can be used 'as is' and will not be further documented. The
Experimental settings WILL change.

Shows if there is a signal on the connector.

Value space of the result returned:

<True/False>

Example:
xStatus Experimental Audio Input Connectors Microphone 1 Activity
*s Experimental Audio Input Connectors Microphone 1 Activity: True
** end

xStatus Experimental Audio Input Connectors Line [1..2] Activity

NOTE: This Experimental command can be used 'as is' and will not be further documented. The
Experimental settings WILL change.

Shows if there is a signal on the connector.

Value space of the result returned:

<True/False>

Example:
xStatus Experimental Audio Input Connectors Line 1 Activity
*s Experimental Audio Input Connectors Line 1 Activity: False
** end

xStatus Experimental Audio Input Connectors HDMI [3, 4] Activity

NOTE: This Experimental command can be used 'as is' and will not be further documented. The
Experimental settings WILL change.

Shows if there is a signal on the connector.

Value space of the result returned:

<True/False>

Example:
xStatus Experimental Audio Input Connectors HDMI 3 Activity
*s Experimental Audio Input Connectors HDMI 3 Activity: False
** end

xStatus Experimental Conference Site [1..n] Capabilities FECC

NOTE: This Experimental command can be used 'as is' and will not be further documented. The
Experimental settings WILL change.

Shows the far end camera control status. This command is available only when in a conference.

Value space of the result returned:

<On/Off>

Example:
xStatus Experimental Conference
*s Experimental Conference Site 18 Capabilities FECC: On
** end

xStatus Experimental Audio StereoEchoCancellation Mode

NOTE: This Experimental command can be used 'as is' and will not be further documented. The
Experimental settings WILL change.

Shows the stereo echo cancellation mode.

Value space of the result returned:

<On/Off>

Example:
xStatus Experimental Audio StereoEchoCancellation Mode
*s Experimental Audio StereoEchoCancellation Mode: Off
** end
| Contents | Introduction | About the API | xConfiguration | xCommand | xStatus | Appendices |

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
```

About 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 key to remove characters in a text field.

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.

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.
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.

<table>
<thead>
<tr>
<th>Button codes - Remote control 5</th>
<th>Button codes - Remote control 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dec</td>
<td>Hex</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>0A</td>
</tr>
<tr>
<td>11</td>
<td>0B</td>
</tr>
<tr>
<td>12</td>
<td>0C</td>
</tr>
<tr>
<td>13</td>
<td>0D</td>
</tr>
<tr>
<td>14</td>
<td>0E</td>
</tr>
<tr>
<td>15</td>
<td>0F</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
</tr>
<tr>
<td>17</td>
<td>11</td>
</tr>
<tr>
<td>18</td>
<td>12</td>
</tr>
<tr>
<td>19</td>
<td>13</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
</tr>
<tr>
<td>21</td>
<td>15</td>
</tr>
<tr>
<td>22</td>
<td>16</td>
</tr>
<tr>
<td>23</td>
<td>17</td>
</tr>
<tr>
<td>24</td>
<td>18</td>
</tr>
<tr>
<td>25</td>
<td>19</td>
</tr>
<tr>
<td>26</td>
<td>1A</td>
</tr>
<tr>
<td>27</td>
<td>1B</td>
</tr>
<tr>
<td>28</td>
<td>1C</td>
</tr>
<tr>
<td>29</td>
<td>1D</td>
</tr>
<tr>
<td>30</td>
<td>1E</td>
</tr>
<tr>
<td>31</td>
<td>1F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IR Signal parameters</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>Siemens SDA2208</td>
<td></td>
</tr>
<tr>
<td>Reference frequency</td>
<td>485 kHz</td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>4 and 7</td>
<td></td>
</tr>
<tr>
<td>IR wavelength</td>
<td>940 nm</td>
<td></td>
</tr>
<tr>
<td>IR carrier frequency</td>
<td>30 kHz</td>
<td></td>
</tr>
</tbody>
</table>
# Overview of the xConfiguration commands

## The Audio settings

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>xConfiguration Audio Input HDMI [2] Level</td>
<td>29</td>
</tr>
<tr>
<td>xConfiguration Audio Input HDMI [2] VideoAssociation MuteOnInactiveVideo</td>
<td>29</td>
</tr>
<tr>
<td>xConfiguration Audio Input Line [1..2] Channel</td>
<td>30</td>
</tr>
<tr>
<td>xConfiguration Audio Input Line [1..2] Equalizer ID</td>
<td>30</td>
</tr>
<tr>
<td>xConfiguration Audio Input Line [1..2] Equalizer Mode</td>
<td>30</td>
</tr>
<tr>
<td>xConfiguration Audio Input Line [1..2] Level</td>
<td>30</td>
</tr>
<tr>
<td>xConfiguration Audio Input Line [1..2] LoopSuppression</td>
<td>30</td>
</tr>
<tr>
<td>xConfiguration Audio Input Line [1..2] Mode</td>
<td>31</td>
</tr>
<tr>
<td>xConfiguration Audio Input Line [1..2] VideoAssociation MuteOnInactiveVideo</td>
<td>30</td>
</tr>
<tr>
<td>xConfiguration Audio Input Line [1..2] VideoAssociation VideoInputSource</td>
<td>30</td>
</tr>
<tr>
<td>xConfiguration Audio Input Microphone [1..2][1..4] EchoControl Mode</td>
<td>31</td>
</tr>
<tr>
<td>xConfiguration Audio Input Microphone [1..2][1..4] Equalizer ID</td>
<td>31</td>
</tr>
<tr>
<td>xConfiguration Audio Input Microphone [1..2][1..4] Equalizer Mode</td>
<td>31</td>
</tr>
<tr>
<td>xConfiguration Audio Input Microphone [1..2][1..4] Level</td>
<td>31</td>
</tr>
<tr>
<td>xConfiguration Audio Input Microphone [1..2][1..4] Mode</td>
<td>32</td>
</tr>
<tr>
<td>xConfiguration Audio Input Microphone [1..2][1..4] VideoAssociation MuteOnInactiveVideo</td>
<td>31</td>
</tr>
<tr>
<td>xConfiguration Audio Input Microphone [1..2][1..4] VideoAssociation VideoInputSource</td>
<td>31</td>
</tr>
<tr>
<td>xConfiguration Audio Input Microphone [1..4] EchoControl Dereverberation</td>
<td>30</td>
</tr>
<tr>
<td>xConfiguration Audio Output HDMI [1] Level</td>
<td>32</td>
</tr>
<tr>
<td>xConfiguration Audio Output HDMI [1] Mode</td>
<td>32</td>
</tr>
<tr>
<td>xConfiguration Audio Output Line [1..2] Channel</td>
<td>32</td>
</tr>
<tr>
<td>xConfiguration Audio Output Line [1..2] Equalizer ID</td>
<td>32</td>
</tr>
<tr>
<td>xConfiguration Audio Output Line [1..2] Equalizer Mode</td>
<td>32</td>
</tr>
<tr>
<td>xConfiguration Audio Output Line [1..2] Level</td>
<td>32</td>
</tr>
<tr>
<td>xConfiguration Audio Output Line [1..2] Mode</td>
<td>33</td>
</tr>
<tr>
<td>xConfiguration Audio Output Line [1..2] Type</td>
<td>33</td>
</tr>
<tr>
<td>xConfiguration Audio Output Line [2] Type</td>
<td>33</td>
</tr>
<tr>
<td>xConfiguration Audio SoundsAndAlerts KeyTones Mode</td>
<td>33</td>
</tr>
<tr>
<td>xConfiguration Audio SoundsAndAlerts RingTone</td>
<td>33</td>
</tr>
<tr>
<td>xConfiguration Audio SoundsAndAlerts RingVolume</td>
<td>33</td>
</tr>
<tr>
<td>xConfiguration Audio Volume</td>
<td>33</td>
</tr>
</tbody>
</table>

## The Cameras settings

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>xConfiguration Cameras Camera [1..7] Backlight</td>
<td>34</td>
</tr>
<tr>
<td>xConfiguration Cameras Camera [1..7] Brightness Level</td>
<td>34</td>
</tr>
<tr>
<td>xConfiguration Cameras Camera [1..7] Brightness Mode</td>
<td>34</td>
</tr>
<tr>
<td>xConfiguration Cameras Camera [1..7] DHCP</td>
<td>35</td>
</tr>
<tr>
<td>xConfiguration Cameras Camera [1..7] Flip</td>
<td>34</td>
</tr>
<tr>
<td>xConfiguration Cameras Camera [1..7] Focus Mode</td>
<td>34</td>
</tr>
<tr>
<td>xConfiguration Cameras Camera [1..7] Gamma Level</td>
<td>35</td>
</tr>
<tr>
<td>xConfiguration Cameras Camera [1..7] Gamma Mode</td>
<td>34</td>
</tr>
<tr>
<td>xConfiguration Cameras Camera [1..7] IrSensor</td>
<td>35</td>
</tr>
<tr>
<td>xConfiguration Cameras Camera [1..7] Mirror</td>
<td>35</td>
</tr>
<tr>
<td>xConfiguration Cameras Camera [1..7] WhiteBalance Level</td>
<td>35</td>
</tr>
<tr>
<td>xConfiguration Cameras Camera [1..7] WhiteBalance Mode</td>
<td>35</td>
</tr>
<tr>
<td>xConfiguration Cameras PowerLine Frequency</td>
<td>34</td>
</tr>
</tbody>
</table>

## The Conference settings

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>xConfiguration Conference [1..1] AutoAnswer Delay</td>
<td>36</td>
</tr>
<tr>
<td>xConfiguration Conference [1..1] AutoAnswer Mode</td>
<td>36</td>
</tr>
<tr>
<td>xConfiguration Conference [1..1] AutoAnswer Mute</td>
<td>36</td>
</tr>
<tr>
<td>xConfiguration Conference [1..1] DefaultCall Protocol</td>
<td>37</td>
</tr>
<tr>
<td>xConfiguration Conference [1..1] DefaultCall Rate</td>
<td>37</td>
</tr>
<tr>
<td>xConfiguration Conference [1..1] DoNotDisturb Mode</td>
<td>36</td>
</tr>
<tr>
<td>xConfiguration Conference [1..1] Encryption Mode</td>
<td>37</td>
</tr>
<tr>
<td>xConfiguration Conference [1..1] FarEndControl Mode</td>
<td>36</td>
</tr>
<tr>
<td>xConfiguration Conference [1..1] FarEndControl SignalCapability</td>
<td>36</td>
</tr>
<tr>
<td>xConfiguration Conference [1..1] IncomingMultisiteCall Mode</td>
<td>36</td>
</tr>
<tr>
<td>xConfiguration Conference [1..1] MaxReceiveCallRate</td>
<td>37</td>
</tr>
<tr>
<td>xConfiguration Conference [1..1] MaxTransmitCallRate</td>
<td>37</td>
</tr>
<tr>
<td>xConfiguration Conference [1..1] MuteUnmuteOnDisconnect</td>
<td>36</td>
</tr>
<tr>
<td>xConfiguration Conference [1..1] PacketLossResilience Mode</td>
<td>38</td>
</tr>
<tr>
<td>xConfiguration Conference [1..1] VideoBandwidth MainChannel Weight</td>
<td>37</td>
</tr>
<tr>
<td>xConfiguration Conference [1..1] VideoBandwidth Mode</td>
<td>37</td>
</tr>
<tr>
<td>xConfiguration Conference [1..1] VideoBandwidth PresentationChannel Weight</td>
<td>37</td>
</tr>
</tbody>
</table>

## The GPIO settings

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>xConfiguration GPIO Pin [1..4] Mode</td>
<td>38</td>
</tr>
</tbody>
</table>
### The H323 settings

- xConfiguration H323 Profile [1..1] Authentication LoginName
- xConfiguration H323 Profile [1..1] Authentication Mode
- xConfiguration H323 Profile [1..1] Authentication Password
- xConfiguration H323 Profile [1..1] CallSetup Mode
- xConfiguration H323 Profile [1..1] Gatekeeper Address
- xConfiguration H323 Profile [1..1] Gatekeeper Discovery
- xConfiguration H323 Profile [1..1] H323Alias E164
- xConfiguration H323 Profile [1..1] H323Alias ID
- xConfiguration H323 Profile [1..1] PortAllocation

### The Network settings

- xConfiguration Network [1..1] Assignment
- xConfiguration Network [1..1] DNS Domain Name
- xConfiguration Network [1..1] DNS Server [1..5] Address
- xConfiguration Network [1..1] IEEE8021X AnonymousIdentity
- xConfiguration Network [1..1] IEEE8021X Eap Md5
- xConfiguration Network [1..1] IEEE8021X Eap Peap
- xConfiguration Network [1..1] IEEE8021X Eap TtLS
- xConfiguration Network [1..1] IEEE8021X Identity
- xConfiguration Network [1..1] IEEE8021X Mode
- xConfiguration Network [1..1] IEEE8021X Password
- xConfiguration Network [1..1] IPStack
- xConfiguration Network [1..1] IPv4 Address
- xConfiguration Network [1..1] IPv4 Gateway
- xConfiguration Network [1..1] IPv4 SubnetMask
- xConfiguration Network [1..1] IPv6 Address
- xConfiguration Network [1..1] IPv6 Assignment
- xConfiguration Network [1..1] IPv6 DHCPOptions
- xConfiguration Network [1..1] IPv6 Gateway
- xConfiguration Network [1..1] MTU
- xConfiguration Network [1..1] QoS Diffserv Audio
- xConfiguration Network [1..1] QoS Diffserv Data
- xConfiguration Network [1..1] QoS Diffserv Signalling
- xConfiguration Network [1..1] QoS Mode
- xConfiguration Network [1..1] RemoteAccess Allow
- xConfiguration Network [1..1] Speed
- xConfiguration Network [1..1] TrafficControl Mode
- xConfiguration Network [1..1] VLAN Voice Mode
- xConfiguration Network [1..1] VLAN Voice Priority
- xConfiguration Network [1..1] VLAN Voice Vid

### The NetworkPort settings


### The NetworkServices settings

- xConfiguration NetworkServices H323 Mode
- xConfiguration NetworkServices HTTP Mode
- xConfiguration NetworkServices HTTPS Mode
- xConfiguration NetworkServices HTTPS VerifyClientCertificate
- xConfiguration NetworkServices HTTPS VerifyServerCertificate
- xConfiguration NetworkServices Multiway Address
- xConfiguration NetworkServices Multiway Protocol
- xConfiguration NetworkServices NTP Address
- xConfiguration NetworkServices NTP Mode
- xConfiguration NetworkServices SIP Mode
- xConfiguration NetworkServices SNMP CommunityName
- xConfiguration NetworkServices SNMP Host [1..3] Address
- xConfiguration NetworkServices SNMP Mode
- xConfiguration NetworkServices SNMP SystemContact
- xConfiguration NetworkServices SNMP SystemLocation
- xConfiguration NetworkServices SSH AllowPublicKey
- xConfiguration NetworkServices SSH Mode
- xConfiguration NetworkServices Telnet Mode

### The Phonebook settings

- xConfiguration Phonebook Server [1..1] ID
- xConfiguration Phonebook Server [1..1] Type
- xConfiguration Phonebook Server [1..1] URL

### The Provisioning settings

- xConfiguration Provisioning ExternalManager Address
- xConfiguration Provisioning ExternalManager Domain
- xConfiguration Provisioning ExternalManager Protocol
- xConfiguration Provisioning HttpMethod
- xConfiguration Provisioning LoginName
- xConfiguration Provisioning Mode
- xConfiguration Provisioning Password

### The Security settings

- xConfiguration Security Audit Logging Mode
- xConfiguration Security Audit OnError Action
- xConfiguration Security Audit Server Address
- xConfiguration Security Audit Server Port
- xConfiguration Security Session InactivityTimeout

### The SerialPort settings

- xConfiguration SerialPort BaudRate
- xConfiguration SerialPort LoginRequired
- xConfiguration SerialPort Mode
### The SIP settings

- xConfiguration SIP Profile [1..1] Authentication [1..1] LoginName
- xConfiguration SIP Profile [1..1] Authentication [1..1] Password
- xConfiguration SIP Profile [1..1] DefaultTransport
- xConfiguration SIP Profile [1..1] Outbound
- xConfiguration SIP Profile [1..1] Proxy [1..4] Address
- xConfiguration SIP Profile [1..1] Proxy [1..4] Discovery
- xConfiguration SIP Profile [1..1] Type
- xConfiguration SIP Profile [1..1] URI

### The Standby settings

- xConfiguration Standby BootAction
- xConfiguration Standby Control
- xConfiguration Standby Delay
- xConfiguration Standby StandbyAction
- xConfiguration Standby WakeupAction

### The SystemUnit settings

- xConfiguration SystemUnit CallLogging Mode
- xConfiguration SystemUnit ContactInfo Type
- xConfiguration SystemUnit IRSensor Mode
- xConfiguration SystemUnit MenuLanguage
- xConfiguration SystemUnit Name

### The Time settings

- xConfiguration Time DateFormat
- xConfiguration Time TimeFormat
- xConfiguration Time Zone

### The Video settings

- xConfiguration Video AllowWebSnapshots
- xConfiguration Video DefaultPresentationSource
- xConfiguration Video Input DVI [3]/[2..3] Type
- xConfiguration Video Input Source [1..3] CameraControl Mode
- xConfiguration Video Input Source [1..3] Name
- xConfiguration Video Input Source [1..3] OptimalDefinition Profile
- xConfiguration Video Input Source [1..3] OptimalDefinition Threshold 60fps
- xConfiguration Video Input Source [1..3] Quality
- xConfiguration Video Input Source [1..3] Type
- xConfiguration Video Input Source 2 Connector
- xConfiguration Video Input Source 3 Connector
- xConfiguration Video Layout LocalLayoutFamily
- xConfiguration Video Layout RemoteLayoutFamily
- xConfiguration Video Layout ScaleToFit
- xConfiguration Video Layout ScaleToFitThreshold
- xConfiguration Video Layout Scaling
- xConfiguration Video MainVideoSource
- xConfiguration Video Monitors
- xConfiguration Video OSD InputMethod Cyrillic
- xConfiguration Video OSD InputMethod InputLanguage
- xConfiguration Video OSD LoginRequired
- xConfiguration Video OSD Mode
- xConfiguration Video OSD MyContactsExpanded
- xConfiguration Video OSD Output
- xConfiguration Video OSD TodaysBookings
- xConfiguration Video Output Composite [3] MonitorRole
- xConfiguration Video Output Composite [3] OverscanLevel
- xConfiguration Video Output Composite [3] Resolution
- xConfiguration Video Output DVI [2] MonitorRole
- xConfiguration Video Output DVI [2] OverscanLevel
- xConfiguration Video Output DVI [2] Resolution
- xConfiguration Video Output HDMI [1] MonitorRole
- xConfiguration Video Output HDMI [1] OverscanLevel
- xConfiguration Video Output HDMI [1] Resolution
- xConfiguration Video Selfview
- xConfiguration Video SelfviewPosition
- xConfiguration Video WallPaper
The Experimental settings

xConfiguration Experimental Audio Input Microphone [1..4] Channel .............................................................. 65
xConfiguration Experimental Audio Input Microphone [1..4] EchoControl HighPassFilter ........................................ 64
xConfiguration Experimental Audio MicrophoneReinforcement AGC ................................................................. 65
xConfiguration Experimental Audio MicrophoneReinforcement Gain ................................................................. 65
xConfiguration Experimental Audio MicrophoneReinforcement Input Microphone [1..4] Mode ....................... 65
xConfiguration Experimental Audio MicrophoneReinforcement Output Line [1..2] Mode ........................................ 65
xConfiguration Experimental Audio Panning MaxAngle .................................................................................... 66
xConfiguration Experimental Audio Panning Mode .......................................................................................... 65
xConfiguration Experimental Audio Panning MonitorLeft ............................................................................... 66
xConfiguration Experimental Audio Panning MonitorRight ........................................................................... 66
xConfiguration Experimental CapiseFilter ........................................................................................................ 64
xConfiguration Experimental Conference [1..1] PacketLossResilience ForwardErrorCorrection .......................... 64
xConfiguration Experimental Conference [1..1] PacketLossResilience RateAdaption ........................................ 64
xConfiguration Experimental CustomSoftbuttons State [1..2] Softbutton [1..5] Type ........................................ 63
xConfiguration Experimental CustomSoftbuttons State [1..2] Softbutton [1..5] Value ........................................ 63
xConfiguration Experimental NetworkServices UPnP Mode ........................................................................... 64
xConfiguration Experimental NetworkServices UPnP Timeout ........................................................................ 64
xConfiguration Experimental SoftwareUpgrade Mode ..................................................................................... 64
xConfiguration Experimental SoftwareUpgrade ServerAddress .................................................................... 64
xConfiguration Experimental SystemUnit Controller Address ........................................................................ 63
xConfiguration Experimental SystemUnit MenuType ........................................................................................ 66
### Overview of the xCommand commands

#### The Audio commands
- `xCommand Audio Equalizer List` .......................................................... 68
- `xCommand Audio Equalizer Update` .................................................... 69
- `xCommand Audio LocalInput Add` ....................................................... 70
- `xCommand Audio LocalInput AddConnector` ........................................ 71
- `xCommand Audio LocalInput Remove` ................................................ 70
- `xCommand Audio LocalInput RemoveConnector` .............................. 71
- `xCommand Audio LocalInput Update` ................................................. 70
- `xCommand Audio LocalOutput Add` .................................................... 71
- `xCommand Audio LocalOutput AddConnector` ..................................... 72
- `xCommand Audio LocalOutput ConnectInput` ...................................... 73
- `xCommand Audio LocalOutput DisconnectInput` ............................... 73
- `xCommand Audio LocalOutput Remove` ............................................. 72
- `xCommand Audio LocalOutput RemoveConnector` ........................... 72
- `xCommand Audio LocalOutput Update` ............................................. 72
- `xCommand Audio LocalOutput UpdateInputGain` ............................. 73
- `xCommand Audio Microphones Mute` .............................................. 69
- `xCommand Audio Microphones Unmute` .......................................... 69
- `xCommand Audio RemoteInput Update` ............................................ 73
- `xCommand Audio RemoteOutput ConnectInput` ............................... 74
- `xCommand Audio RemoteOutput DisconnectInput` ............................ 74
- `xCommand Audio RemoteOutput UpdateInputGain` .......................... 74
- `xCommand Audio Setup Clear` ......................................................... 74
- `xCommand Audio Sound Play` .......................................................... 74
- `xCommand Audio Sound Stop` .......................................................... 75
- `xCommand Audio VuMeter Start` ....................................................... 75
- `xCommand Audio VuMeter Stop` ....................................................... 75

#### The Call commands
- `xCommand Call Accept` ................................................................. 76
- `xCommand Call Disconnect` ............................................................ 76
- `xCommand Call DisconnectAll` ....................................................... 76
- `xCommand Call Hold` ................................................................. 76
- `xCommand Call Join` ................................................................. 76
- `xCommand Call Reject` ................................................................. 76
- `xCommand Call Resume` ................................................................. 77

#### The CallLog commands
- `xCommand CallLog Clear` ................................................................. 77
- `xCommand CallLog Missed Delete` .................................................. 78
- `xCommand CallLog Missed Reviewed` ............................................. 78
- `xCommand CallLog Outgoing Delete` .............................................. 77
- `xCommand CallLog Received Delete` .............................................. 78
- `xCommand CallLog Recent Delete` ................................................ 77

#### The Camera commands
- `xCommand Camera PanTiltReset` .................................................... 79
- `xCommand Camera PositionActivateFromPreset` ............................ 80
- `xCommand Camera PositionReset` ................................................ 79
- `xCommand Camera PositionSet` ...................................................... 79
- `xCommand Camera Ramp` ............................................................. 80
- `xCommand Camera ReconfigureCameraChain` ............................... 80
- `xCommand Camera TriggerAutofocus` .......................................... 80

#### The CamCtrlPip command
- `xCommand CamCtrlPip` ................................................................. 78

#### The CallLog commands
- `xCommand CallLog Clear` ................................................................. 77
- `xCommand CallLog Missed Delete` .................................................. 78
- `xCommand CallLog Missed Reviewed` ............................................. 78
- `xCommand CallLog Outgoing Delete` .............................................. 77
- `xCommand CallLog Received Delete` .............................................. 78
- `xCommand CallLog Recent Delete` ................................................ 77

#### The Dial command
- `xCommand Dial` ................................................................. 81

#### The DTMFSend command
- `xCommand DTMFSend` ................................................................. 81

#### The FarEndControl commands
- `xCommand FarEndControl Camera Move` ........................................ 82
- `xCommand FarEndControl Camera Stop` ........................................ 82
- `xCommand FarEndControl Preset Activate` .................................... 82
- `xCommand FarEndControl Preset Store` ........................................ 82
- `xCommand FarEndControl Source Select` ....................................... 83

#### The GPIO command
- `xCommand GPIO ManualState Set` ................................................. 83

#### The HttpFeedback commands
- `xCommand HttpFeedback Deregister` ............................................. 84
- `xCommand HttpFeedback Register` ................................................ 84
## Contents

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Key commands</td>
<td>84</td>
</tr>
<tr>
<td>xCommand Key Click</td>
<td>84</td>
</tr>
<tr>
<td>xCommand Key Press</td>
<td>84</td>
</tr>
<tr>
<td>xCommand Key Release</td>
<td>84</td>
</tr>
<tr>
<td>The Message commands</td>
<td>85</td>
</tr>
<tr>
<td>xCommand Message Alert Clear</td>
<td>85</td>
</tr>
<tr>
<td>xCommand Message Alert Display</td>
<td>85</td>
</tr>
<tr>
<td>xCommand Message Echo</td>
<td>87</td>
</tr>
<tr>
<td>xCommand Message Prompt Clear</td>
<td>86</td>
</tr>
<tr>
<td>xCommand Message Prompt Display</td>
<td>86</td>
</tr>
<tr>
<td>xCommand Message Prompt Response</td>
<td>86</td>
</tr>
<tr>
<td>xCommand Message TextLine Clear</td>
<td>85</td>
</tr>
<tr>
<td>xCommand Message TextLine Display</td>
<td>85</td>
</tr>
<tr>
<td>The Phonebook commands</td>
<td>87</td>
</tr>
<tr>
<td>xCommand Phonebook Contact Add</td>
<td>88</td>
</tr>
<tr>
<td>xCommand Phonebook Contact Delete</td>
<td>89</td>
</tr>
<tr>
<td>xCommand Phonebook ContactMethod Add</td>
<td>89</td>
</tr>
<tr>
<td>xCommand Phonebook ContactMethod Delete</td>
<td>89</td>
</tr>
<tr>
<td>xCommand Phonebook Contact Modify</td>
<td>89</td>
</tr>
<tr>
<td>xCommand Phonebook Folder Add</td>
<td>87</td>
</tr>
<tr>
<td>xCommand Phonebook Folder Delete</td>
<td>88</td>
</tr>
<tr>
<td>xCommand Phonebook Folder Modify</td>
<td>87</td>
</tr>
<tr>
<td>xCommand Phonebook Search</td>
<td>90</td>
</tr>
<tr>
<td>The Presentation commands</td>
<td>91</td>
</tr>
<tr>
<td>xCommand Presentation Start</td>
<td>91</td>
</tr>
<tr>
<td>xCommand Presentation Stop</td>
<td>91</td>
</tr>
<tr>
<td>The Preset commands</td>
<td>91</td>
</tr>
<tr>
<td>xCommand Preset Activate</td>
<td>91</td>
</tr>
<tr>
<td>xCommand Preset Clear</td>
<td>92</td>
</tr>
<tr>
<td>xCommand Preset Store</td>
<td>91</td>
</tr>
<tr>
<td>The Provisioning commands</td>
<td>92</td>
</tr>
<tr>
<td>xCommand Provisioning CancelUpgrade</td>
<td>92</td>
</tr>
<tr>
<td>xCommand Provisioning CompleteUpgrade</td>
<td>92</td>
</tr>
<tr>
<td>xCommand Provisioning StartUpgrade</td>
<td>92</td>
</tr>
<tr>
<td>The SStringSend command</td>
<td>93</td>
</tr>
<tr>
<td>xCommand SStringSend</td>
<td>93</td>
</tr>
<tr>
<td>The Standby commands</td>
<td>93</td>
</tr>
<tr>
<td>xCommand Standby Activate</td>
<td>93</td>
</tr>
<tr>
<td>xCommand Standby Deactivate</td>
<td>93</td>
</tr>
<tr>
<td>xCommand Standby ResetTimer</td>
<td>93</td>
</tr>
<tr>
<td>The SystemUnit commands</td>
<td>94</td>
</tr>
<tr>
<td>xCommand SystemUnit AdminPassword Set</td>
<td>94</td>
</tr>
<tr>
<td>xCommand SystemUnit ConfigurationProfile CancelChange</td>
<td>96</td>
</tr>
<tr>
<td>xCommand SystemUnit ConfigurationProfile Change</td>
<td>95</td>
</tr>
<tr>
<td>xCommand SystemUnit ConfigurationProfile List</td>
<td>96</td>
</tr>
<tr>
<td>xCommand SystemUnit ConfigurationProfile Remove</td>
<td>95</td>
</tr>
<tr>
<td>xCommand SystemUnit ConfigurationProfile SaveCurrentConfigurationAS</td>
<td>95</td>
</tr>
<tr>
<td>xCommand SystemUnit DateTime Get</td>
<td>95</td>
</tr>
<tr>
<td>xCommand SystemUnit DateTime Set</td>
<td>94</td>
</tr>
<tr>
<td>xCommand SystemUnit FactoryReset</td>
<td>95</td>
</tr>
<tr>
<td>xCommand SystemUnit MenuPassword Set</td>
<td>94</td>
</tr>
<tr>
<td>xCommand SystemUnit MenuPassword Validate</td>
<td>94</td>
</tr>
<tr>
<td>xCommand SystemUnit OptionKey Add</td>
<td>94</td>
</tr>
<tr>
<td>xCommand SystemUnit ReleaseKey Add</td>
<td>94</td>
</tr>
<tr>
<td>xCommand SystemUnit SoftwareUpgrade</td>
<td>95</td>
</tr>
<tr>
<td>The TStringSend command</td>
<td>96</td>
</tr>
<tr>
<td>xCommand TStringSend</td>
<td>96</td>
</tr>
<tr>
<td>The Video commands</td>
<td>97</td>
</tr>
<tr>
<td>xCommand Video Layout Add</td>
<td>97</td>
</tr>
<tr>
<td>xCommand Video Layout Assign</td>
<td>100</td>
</tr>
<tr>
<td>xCommand Video Layout AssignCall</td>
<td>100</td>
</tr>
<tr>
<td>xCommand Video Layout AssignLocalOutput</td>
<td>100</td>
</tr>
<tr>
<td>xCommand Video Layout AssignPresentation</td>
<td>101</td>
</tr>
<tr>
<td>xCommand Video Layout Frame Add</td>
<td>98</td>
</tr>
<tr>
<td>xCommand Video Layout Frame Remove</td>
<td>99</td>
</tr>
<tr>
<td>xCommand Video Layout Frame Update</td>
<td>99</td>
</tr>
<tr>
<td>xCommand Video Layout ListLayoutFamily</td>
<td>97</td>
</tr>
<tr>
<td>xCommand Video Layout ListLayoutGraphic</td>
<td>97</td>
</tr>
<tr>
<td>xCommand Video Layout LoadDB</td>
<td>102</td>
</tr>
<tr>
<td>xCommand Video Layout Remove</td>
<td>98</td>
</tr>
<tr>
<td>xCommand Video Layout RemoveAll</td>
<td>98</td>
</tr>
<tr>
<td>xCommand Video Layout Reset</td>
<td>98</td>
</tr>
<tr>
<td>xCommand Video Layout SetPresentationView</td>
<td>102</td>
</tr>
<tr>
<td>xCommand Video Layout UnAssign</td>
<td>101</td>
</tr>
<tr>
<td>xCommand Video Layout UnAssignCall</td>
<td>101</td>
</tr>
<tr>
<td>xCommand Video Layout UnAssignLocalOutput</td>
<td>101</td>
</tr>
<tr>
<td>xCommand Video Layout UnAssignPresentation</td>
<td>102</td>
</tr>
<tr>
<td>xCommand Video PictureLayoutSet</td>
<td>97</td>
</tr>
<tr>
<td>The Experimental commands</td>
<td>xCommand Experimental Audio LocalInput Mute</td>
</tr>
</tbody>
</table>
Overview of the xStatus commands

The Audio status .................................................................................................................. 107
  xStatus Audio .................................................................................................................. 107
  xStatus Audio Input LocalInput [1..n] AGC ................................................................. 109
  xStatus Audio Input LocalInput [1..n] Channels .......................................................... 109
  xStatus Audio Input LocalInput [1..n] Connector ........................................................ 109
  xStatus Audio Input LocalInput [1..n] MixerMode ......................................................... 108
  xStatus Audio Input LocalInput [1..n] Mute .................................................................... 108
  xStatus Audio Input LocalInput [1..n] Name .................................................................. 108
  xStatus Audio Input RemoteInput [1..n] AGC .............................................................. 109
  xStatus Audio Input RemoteInput [1..n] CallId ............................................................... 109
  xStatus Audio Microphones Mute ................................................................................... 108
  xStatus Audio Module [1..n] Connector ........................................................................ 111
  xStatus Audio Module [1..n] HardwareID ....................................................................... 111
  xStatus Audio Module [1..n] SoftwareID ........................................................................ 111
  xStatus Audio Module [1..n] Type .................................................................................. 110
  xStatus Audio Output LocalOutput [1..n] Channels ...................................................... 110
  xStatus Audio Output LocalOutput [1..n] Connector ...................................................... 110
  xStatus Audio Output LocalOutput [1..n] Input [1..n] Gain ................................................ 110
  xStatus Audio Output LocalOutput [1..n] Loudspeaker ................................................ 109
  xStatus Audio Output RemoteOutput [1..n] CallId ........................................................ 110
  xStatus Audio Output RemoteOutput [1..n] Input [1..n] Gain ............................................. 110
  xStatus Audio Volume ..................................................................................................... 108

The Call status ...................................................................................................................... 111
  xStatus Call .................................................................................................................... 111
  xStatus Call [1..n] CallbackNumber .............................................................................. 112
  xStatus Call [1..n] CallType ............................................................................................ 112
  xStatus Call [1..n] Direction ........................................................................................... 111
  xStatus Call [1..n] DisplayName ...................................................................................... 112
  xStatus Call [1..n] Duration ............................................................................................ 113
  xStatus Call [1..n] Encryption Type ............................................................................... 112
  xStatus Call [1..n] PlacedOnHold .................................................................................... 113
  xStatus Call [1..n] Protocol ............................................................................................ 112
  xStatus Call [1..n] ReceiveCallRate ................................................................................. 112
  xStatus Call [1..n] RemoteNumber .................................................................................. 112
  xStatus Call [1..n] Status ................................................................................................ 111
  xStatus Call [1..n] TransmitCallRate ............................................................................. 112

The Camera status ............................................................................................................... 113
  xStatus Camera .............................................................................................................. 113
  xStatus Camera [1..7] Capabilities Options ................................................................. 115
  xStatus Camera [1..7] Connected .................................................................................. 114
  xStatus Camera [1..7] HardwareID ................................................................................. 114
  xStatus Camera [1..7] IpAddress .................................................................................... 114
  xStatus Camera [1..7] MacAddress ................................................................................. 115
  xStatus Camera [1..7] Manufacturer ............................................................................. 114
  xStatus Camera [1..7] Model ........................................................................................ 114
  xStatus Camera [1..7] Position Focus ............................................................................ 115
  xStatus Camera [1..7] Position Pan ............................................................................... 115
  xStatus Camera [1..7] Position Tilt ............................................................................... 115
  xStatus Camera [1..7] Position Zoom ........................................................................... 115
  xStatus Camera [1..7] SerialNumber ............................................................................ 114
  xStatus Camera [1..7] SoftwareID ................................................................................ 114

The Conference status ........................................................................................................ 116
  xStatus Conference ........................................................................................................ 116
  xStatus Conference Presentation LocalSource ............................................................. 116
  xStatus Conference Presentation Mode .......................................................................... 116
  xStatus Conference Presentation Protocol ...................................................................... 116
  xStatus Conference Presentation Resolution Height ................................................... 116
  xStatus Conference Presentation Resolution Width ..................................................... 116
  xStatus Conference Presentation SiteId ........................................................................ 116
  xStatus Conference Site [1..n] Capabilities Presentation .............................................. 117
  xStatus Conference Site [1..n] MicrophonesMuted ...................................................... 117
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Network status</td>
<td>127</td>
</tr>
<tr>
<td>xStatus Network</td>
<td>127</td>
</tr>
<tr>
<td>xStatus Network 1 Ethernet MacAddress</td>
<td>127</td>
</tr>
<tr>
<td>xStatus Network 1 Ethernet Speed</td>
<td>127</td>
</tr>
<tr>
<td>xStatus Network 1 IPv4 Address</td>
<td>128</td>
</tr>
<tr>
<td>xStatus Network 1 IPv4 DNS Domain Name</td>
<td>128</td>
</tr>
<tr>
<td>xStatus Network 1 IPv4 DNS Server [1..5] Address</td>
<td>128</td>
</tr>
<tr>
<td>xStatus Network 1 IPv4 Gateway</td>
<td>128</td>
</tr>
<tr>
<td>xStatus Network 1 IPv6 Address</td>
<td>128</td>
</tr>
<tr>
<td>xStatus Network 1 IPv6 Gateway</td>
<td>128</td>
</tr>
<tr>
<td>xStatus Network 1 MTU</td>
<td>128</td>
</tr>
<tr>
<td>The Preset status</td>
<td>129</td>
</tr>
<tr>
<td>xStatus Preset</td>
<td>129</td>
</tr>
<tr>
<td>xStatus Preset [1..15] Defined</td>
<td>129</td>
</tr>
<tr>
<td>xStatus Preset [1..15] Description</td>
<td>129</td>
</tr>
<tr>
<td>xStatus Preset [1..15] Type</td>
<td>129</td>
</tr>
<tr>
<td>The Provisioning status</td>
<td>130</td>
</tr>
<tr>
<td>xStatus Provisioning</td>
<td>130</td>
</tr>
<tr>
<td>xStatus Provisioning PendingUpgrade</td>
<td>130</td>
</tr>
<tr>
<td>xStatus Provisioning Reason</td>
<td>130</td>
</tr>
<tr>
<td>xStatus Provisioning Status</td>
<td>130</td>
</tr>
<tr>
<td>The SIP status</td>
<td>131</td>
</tr>
<tr>
<td>xStatus SIP</td>
<td>131</td>
</tr>
<tr>
<td>xStatus SIP Profile 1 Authentication</td>
<td>133</td>
</tr>
<tr>
<td>xStatus SIP Profile 1 Proxy [1..4] Address</td>
<td>133</td>
</tr>
<tr>
<td>xStatus SIP Profile 1 Proxy [1..4] Status</td>
<td>132</td>
</tr>
<tr>
<td>xStatus SIP Profile 1 Registration [1..n] Reason</td>
<td>133</td>
</tr>
<tr>
<td>xStatus SIP Profile 1 Registration [1..n] Status</td>
<td>133</td>
</tr>
<tr>
<td>xStatus SIP Profile 1 Registration [1..n] URI</td>
<td>133</td>
</tr>
<tr>
<td>xStatus SIP Profile 1 Secure</td>
<td>133</td>
</tr>
<tr>
<td>xStatus SIP Profile 1 Verified</td>
<td>133</td>
</tr>
<tr>
<td>xStatus SIP Proxy 1 Address</td>
<td>131</td>
</tr>
<tr>
<td>xStatus SIP Proxy 1 Secure</td>
<td>131</td>
</tr>
<tr>
<td>xStatus SIP Proxy 1 Status</td>
<td>131</td>
</tr>
<tr>
<td>xStatus SIP Proxy 1 Verified</td>
<td>132</td>
</tr>
<tr>
<td>xStatus SIP Registration [1..n] Authentication</td>
<td>132</td>
</tr>
<tr>
<td>xStatus SIP Registration [1..n] Reason</td>
<td>132</td>
</tr>
<tr>
<td>xStatus SIP Registration [1..n] Status</td>
<td>132</td>
</tr>
<tr>
<td>xStatus SIP Registration [1..n] URI</td>
<td>132</td>
</tr>
<tr>
<td>The Sleep status</td>
<td>134</td>
</tr>
<tr>
<td>xStatus Sleep Active</td>
<td>134</td>
</tr>
<tr>
<td>The Standby status</td>
<td>134</td>
</tr>
<tr>
<td>xStatus Standby Active</td>
<td>134</td>
</tr>
<tr>
<td>The SystemUnit status</td>
<td>135</td>
</tr>
<tr>
<td>xStatus SystemUnit</td>
<td>135</td>
</tr>
<tr>
<td>xStatus SystemUnit ContactInfo</td>
<td>139</td>
</tr>
<tr>
<td>xStatus SystemUnit Hardware AudioBoard Identifier</td>
<td>139</td>
</tr>
<tr>
<td>xStatus SystemUnit Hardware AudioBoard SerialNumber</td>
<td>138</td>
</tr>
<tr>
<td>xStatus SystemUnit Hardware BootSoftware</td>
<td>138</td>
</tr>
<tr>
<td>xStatus SystemUnit Hardware MainBoard Identifier</td>
<td>137</td>
</tr>
<tr>
<td>xStatus SystemUnit Hardware MainBoard SerialNumber</td>
<td>137</td>
</tr>
<tr>
<td>xStatus SystemUnit Software Name</td>
<td>136</td>
</tr>
<tr>
<td>xStatus SystemUnit Software OptionKeys MultiSite</td>
<td>136</td>
</tr>
<tr>
<td>xStatus SystemUnit Software OptionKeys NaturalPresenter</td>
<td>136</td>
</tr>
<tr>
<td>xStatus SystemUnit Software OptionKeys PremiumResolution</td>
<td>137</td>
</tr>
<tr>
<td>xStatus SystemUnit Software ReleaseDate</td>
<td>136</td>
</tr>
<tr>
<td>xStatus SystemUnit Software ReleaseKey</td>
<td>136</td>
</tr>
<tr>
<td>xStatus SystemUnit Software Version</td>
<td>136</td>
</tr>
<tr>
<td>xStatus SystemUnit State MaxNumberOfActiveCalls</td>
<td>138</td>
</tr>
<tr>
<td>xStatus SystemUnit State MaxNumberOfCalls</td>
<td>138</td>
</tr>
<tr>
<td>xStatus SystemUnit State NumberOfActiveCalls</td>
<td>139</td>
</tr>
<tr>
<td>xStatus SystemUnit State NumberOfInProgressCalls</td>
<td>139</td>
</tr>
<tr>
<td>xStatus SystemUnit State NumberOfSuspendedCalls</td>
<td>139</td>
</tr>
<tr>
<td>xStatus SystemUnit State Subsystem Application</td>
<td>139</td>
</tr>
<tr>
<td>xStatus SystemUnit State System</td>
<td>138</td>
</tr>
<tr>
<td>xStatus SystemUnit Uptime</td>
<td>135</td>
</tr>
<tr>
<td>The Time status</td>
<td>140</td>
</tr>
<tr>
<td>xStatus Time Zone Olson</td>
<td>140</td>
</tr>
</tbody>
</table>
### The Video Input status
- `xStatus Video Input [1..n] Device [1..3] Connected`...
- `xStatus Video Input DVI [2..3] Signal State`...
- `xStatus Video Input DVI [2..3] Signal State`...
- `xStatus Video Input HDMI [1..2] Connected`...
- `xStatus Video Input Legacy [3] Connected`...
- `xStatus Video Input Legacy [3] Signal State`...
- `xStatus Video Input Site [1..n] Output [1..3] Resolution Refresh Rate`...
- `xStatus Video Input Source [1..3] Resolution Refresh Rate`...

### The Video Layout status
- `xStatus Video Layout Site [1..n] Output [1..3] Descriptor Output`...
- `xStatus Video Layout Site [1..n] Output [1..3] Descriptor Output`...
- `xStatus Video Layout Site [1..n] Output [1..3] Family Name`...
- `xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] Height`...
- `xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] Input Number`...
- `xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] Layer`...
- `xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] Position X`...
- `xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] Position Y`...
- `xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] Video Source Type`...
- `xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] Width`...
- `xStatus Video Layout Site [1..n] Output [1..3] Full Family Name`...
- `xStatus Video Layout Site [1..n] Output [1..3] Graphic Name`...
- `xStatus Video Layout Site [1..n] Output [1..3] Graphic Number`...

### The Video Output status
- `xStatus Video Output [1..n] Device [1..3] Resolution Height`...
- `xStatus Video Output DVI [2] Resolution Height`...
- `xStatus Video Output HDMI [1] Resolution Height`...
- `xStatus Video Output Legacy [3] Resolution Width`...
- `xStatus Video Output [1..n] Frame [1..6] Position X`...
- `xStatus Video Output [1..n] Frame [1..6] Position Y`...
- `xStatus Video Output [1..n] Frame [1..6] Video Source Type`...

### The Experimental status
- `xStatus Experimental Audio Input Connectors HDMI [3, 4] Activity`...
- `xStatus Experimental Audio Input Connectors Line [1..2] Activity`...
- `xStatus Experimental Audio Input Connectors Microphone [1..4] Activity`...
- `xStatus Experimental Software Stereo ECHO Cancellation Mode`...
- `xStatus Experimental Conference Site Capabilities FECC`...
- `xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Packet Recovery Media Packets Consecutive Lost`...
- `xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Packet Recovery Media Packets Lost`...
- `xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Packet Recovery Media Packets Out Of Order`...
- `xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Packet Recovery Media Packets Received`...
- `xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Packet Recovery Media Packets Recovered`...
- `xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Packet Recovery Media Packets Statistics More Than Four Lost`...
- `xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Packet Recovery Media Packets Statistics One Lost`...
- `xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Packet Recovery Media Packets Statistics Three Lost`...
- `xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Packet Recovery Media Packets Statistics Two Lost`...
- `xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Packet Recovery Protection Packets Lost`...
- `xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Packet Recovery Protection Packets Media Packets Covered`...
- `xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Packet Recovery Protection Packets Out Of Order`...
- `xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Packet Recovery Protection Packets Received`...
- `xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Packet Recovery Media Packets Consecutive Lost`...
- `xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Packet Recovery Media Packets Lost`...
- `xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Packet Recovery Media Packets Recovered`...
- `xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Packet Recovery Media Packets Statistics Four Lost`...
- `xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Packet Recovery Media Packets Statistics One Lost`...
- `xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Packet Recovery Media Packets Statistics More Than Four Lost`...
- `xStatus Experimental Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Packet Recovery Media Packets Statistics One Lost`...
<table>
<thead>
<tr>
<th>Contents</th>
<th>Introduction</th>
<th>About the API</th>
<th>xConfiguration</th>
<th>xCommand</th>
<th>xStatus</th>
<th>Appendices</th>
</tr>
</thead>
<tbody>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] PacketRecovery MediaPackets Statistics ThreeLost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] PacketRecovery MediaPackets Statistics TwoLost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] PacketRecovery MediaPackets Statistics ZeroLost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] PacketRecovery ProtectionPackets Lost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] PacketRecovery ProtectionPackets MediaPacketsCovered</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] PacketRecovery ProtectionPackets OutOfOrder</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] PacketRecovery ProtectionPackets Received</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] PacketRecovery MediaPackets ConsecutiveLost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] PacketRecovery MediaPackets Lost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] PacketRecovery MediaPackets OutOfOrder</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] PacketRecovery MediaPackets Received</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] PacketRecovery MediaPackets Recovered</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] PacketRecovery MediaPackets Statistics FourLost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] PacketRecovery MediaPackets Statistics MoreThanFourLost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] PacketRecovery MediaPackets Statistics OneLost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] PacketRecovery MediaPackets Statistics ThreeLost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] PacketRecovery MediaPackets Statistics TwoLost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] PacketRecovery MediaPackets Statistics ZeroLost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] PacketRecovery ProtectionPackets Lost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] PacketRecovery ProtectionPackets MediaPacketsCovered</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] PacketRecovery ProtectionPackets OutOfOrder</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xStatus Experimental Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] PacketRecovery ProtectionPackets Received</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
On our web site you will find an overview of the worldwide Cisco contacts.
Go to:  [http://www.cisco.com/web/siteassets/contacts](http://www.cisco.com/web/siteassets/contacts)

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA 95134 USA

Disclaimer

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks.
Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

TANDBERG is now a part of Cisco. TANDBERG® is a registered trademark belonging to Tandberg ASA.