Software version TC3.0
MARCH 2010

TANDBERG API Guide

TANDBERG Codec C60 and Codec 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 the TANDBERG web site regularly for updated versions of this guide. Go to: http://www.tandberg.com/docs

Table of Contents

Introduction

Introduction ...............................................................
The purpose of this document .................................
Disclaimers and Notices .........................................
Products covered in this guide ............................... Other documents you might find useful ..............
TANDBERG Codec C40 ...........................................
What’s new in this version ...................................... Software release notes ......................................
User documentation ..............................................
New commands .....................................................
xConfiguration .......................................................
xCommand ........................................................
xStatus ................................................................
Commands that have been changed .................
xConfiguration .......................................................
xCommand ........................................................
xStatus ................................................................
Commands that are removed ..........................
xConfiguration ....................................................... xCommand ........................................................

About the TANDBERG API

TANDBERG API .....................................................
Basic Principles ..................................................
The TANDBERG API-Engine .............................. Structuring of Information ............................... Addressing Using XPath or TANDBERG SimplePath Feedback ..................................................
Connecting to the codec .................................... Accessing XACLI ...........................................
Telnet/SSH login ..................................................
Serial port login ..................................................
Serial port configurations ..................................
Reboot .............................................................. Hardware & Cabling (RS-232) ......................... Troubleshooting (RS-232) ...............................
### Description of the xConfiguration commands

<table>
<thead>
<tr>
<th>Description of the xConfiguration commands</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Audio settings</td>
<td>27</td>
</tr>
<tr>
<td>The Camera settings</td>
<td>31</td>
</tr>
<tr>
<td>The Conference settings</td>
<td>33</td>
</tr>
<tr>
<td>The GPIO settings</td>
<td>35</td>
</tr>
<tr>
<td>The H323 settings</td>
<td>35</td>
</tr>
<tr>
<td>The Network settings</td>
<td>37</td>
</tr>
<tr>
<td>The NetworkServices settings</td>
<td>40</td>
</tr>
<tr>
<td>The Phonebook settings</td>
<td>41</td>
</tr>
<tr>
<td>The Provisioning settings</td>
<td>42</td>
</tr>
<tr>
<td>The SerialPort settings</td>
<td>42</td>
</tr>
<tr>
<td>The SIP settings</td>
<td>43</td>
</tr>
<tr>
<td>The Standby settings</td>
<td>44</td>
</tr>
<tr>
<td>The SystemUnit settings</td>
<td>45</td>
</tr>
<tr>
<td>The Time settings</td>
<td>45</td>
</tr>
<tr>
<td>The Video settings</td>
<td>46</td>
</tr>
<tr>
<td>The Experimental menu</td>
<td>52</td>
</tr>
</tbody>
</table>

### Description of the xCommand commands

<table>
<thead>
<tr>
<th>xCommands with parameters</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Audio commands</td>
<td>54</td>
</tr>
<tr>
<td>The Boot command</td>
<td>60</td>
</tr>
<tr>
<td>The Call commands</td>
<td>61</td>
</tr>
<tr>
<td>The Call Log commands</td>
<td>62</td>
</tr>
<tr>
<td>The CamCtrlPIP command</td>
<td>63</td>
</tr>
<tr>
<td>The Camera commands</td>
<td>63</td>
</tr>
<tr>
<td>The Dial command</td>
<td>64</td>
</tr>
<tr>
<td>The DTMF command</td>
<td>65</td>
</tr>
<tr>
<td>The Far End Control commands</td>
<td>65</td>
</tr>
<tr>
<td>The GPIO commands</td>
<td>66</td>
</tr>
<tr>
<td>The HTTP Feedback commands</td>
<td>67</td>
</tr>
<tr>
<td>The Key commands</td>
<td>67</td>
</tr>
<tr>
<td>The Message command</td>
<td>68</td>
</tr>
<tr>
<td>The Phonebook commands</td>
<td>69</td>
</tr>
<tr>
<td>The Presentation commands</td>
<td>71</td>
</tr>
<tr>
<td>The Preset commands</td>
<td>72</td>
</tr>
<tr>
<td>The SString Send command</td>
<td>72</td>
</tr>
<tr>
<td>The Standby command</td>
<td>73</td>
</tr>
<tr>
<td>The SystemUnit command</td>
<td>73</td>
</tr>
</tbody>
</table>

### Description of the xStatus commands

<table>
<thead>
<tr>
<th>xStatus commands</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Audio status</td>
<td>83</td>
</tr>
<tr>
<td>The Call status</td>
<td>87</td>
</tr>
<tr>
<td>The Camera status</td>
<td>89</td>
</tr>
<tr>
<td>The Conference status</td>
<td>91</td>
</tr>
<tr>
<td>The Diagnostics status</td>
<td>93</td>
</tr>
<tr>
<td>The GPIO status</td>
<td>101</td>
</tr>
<tr>
<td>The H323 status</td>
<td>101</td>
</tr>
<tr>
<td>The HTTP Feedback status</td>
<td>102</td>
</tr>
<tr>
<td>The Media Channel status</td>
<td>103</td>
</tr>
<tr>
<td>The Network status</td>
<td>106</td>
</tr>
<tr>
<td>The Preset status</td>
<td>108</td>
</tr>
<tr>
<td>The SIP status</td>
<td>109</td>
</tr>
<tr>
<td>The Standby status</td>
<td>112</td>
</tr>
<tr>
<td>The System Unit status</td>
<td>112</td>
</tr>
<tr>
<td>The Video Input status</td>
<td>117</td>
</tr>
<tr>
<td>The Video Output status</td>
<td>118</td>
</tr>
<tr>
<td>The Video Layout status</td>
<td>120</td>
</tr>
<tr>
<td>The Video commands</td>
<td>76</td>
</tr>
<tr>
<td>The Experimental status commands</td>
<td>123</td>
</tr>
</tbody>
</table>

### Appendices

<table>
<thead>
<tr>
<th>Appendices</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic audio API</td>
<td>125</td>
</tr>
<tr>
<td>The equalizer</td>
<td>126</td>
</tr>
<tr>
<td>The Audio Console application</td>
<td>126</td>
</tr>
<tr>
<td>The equalizer filter parameters</td>
<td>126</td>
</tr>
<tr>
<td>The equalizer IIR filter</td>
<td>126</td>
</tr>
<tr>
<td>The filter types</td>
<td>126</td>
</tr>
<tr>
<td>Stereo</td>
<td>127</td>
</tr>
<tr>
<td>Stereo in point to point call</td>
<td>127</td>
</tr>
<tr>
<td>Stereo in Multisite</td>
<td>127</td>
</tr>
<tr>
<td>Example of local stereo configuration</td>
<td>127</td>
</tr>
<tr>
<td>Startup script</td>
<td>128</td>
</tr>
<tr>
<td>Adding a startup script</td>
<td>128</td>
</tr>
</tbody>
</table>

Remote Control TRC5—Key map ................................................. 129
Remote Control TRC5—One page overview ................................. 130
Chapter 1

Introduction
Introduction

You may already be familiar with the system integrator guides for the Codec C Series. To help you find the information you need, TANDBERG has split the guide into two smaller and more accessible parts listed on the right of this page.

The purpose of this document

The purpose of this document is to introduce you to the TANDBERG Application Programmer Interface (API) in general and to serve as a reference guide for the API commands.

Disclaimers and Notices

The objective of this documentation is to provide the reader with assistance in using and configuring the product. The capabilities of TANDBERG products and other manufacturers’ products change over time and so the required configuration may be different from that indicated here. If you have any suggestions for changes to this document, please feed them back to TANDBERG through your TANDBERG Authorized Service Representative.

If you need technical support, please contact your TANDBERG Authorized Service Representative.

The specifications for the product and the information in this Guide are subject to change at any time, without notice, by TANDBERG. Every effort has been made to supply complete and accurate information in this Guide; however, TANDBERG assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

TANDBERG® is a registered trademark belonging to Tandberg ASA. Other trademarks used in this document are the property of their respective holders.

This Guide may be reproduced in its entirety, including all copyright and intellectual property notices, in limited quantities in connection with the use of this product. Except for the limited exception set forth in the previous sentence, no part of this Guide may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronically, mechanically, by photocopying, or otherwise, without the prior written permission of TANDBERG.

www.tandberg.com

© 2010 TANDBERG

Products covered in this guide

- TANDBERG Codec C60
- TANDBERG Codec C40

Other documents you might find useful

- Video conference room primer guide
- Getting started guide for Profile Series, Codec C Series, Quick Set C20
- User Guide for Profile Series, Codec C Series and Quick Set C20
- Administrator guide for Profiles using Codec C20 and Quick Set C20
- Administrator guide for Profiles using Codec C60 and Codec C60/C40
- Administrator guide for Profiles using Codec C90 and Codec C90
- Camera user guide for PrecisionHD 1080p/720p
- API guides for Codec C90, C60, C40
- Physical interfaces guides for Codec C90, C60, C40
- Legal and safety information for Profile Series, Codec C Series and Quick Set C20

Download the user documentation

Go to: ► http://www.tandberg.com/docs
TANDBERG Codec C40

We are glad to introduce a new codec with the same design and technology principles as the Codec C90 and Codec C60. Supported with software version TC3.0.

The TANDBERG Codec C40 provides all the power required to transform any conference room to a HD video collaboration room. Designed for any standard HD integration project, the Codec C40 is the ideal solution for everyday video conferencing and collaboration solution. 1080p HD video, and Multisite™ features combine to make the Codec C40 ideal for a variety of applications.

The Codec C40 is ideal for standard meeting rooms, executive offices and team collaboration rooms.

- Full High Definition Video with up to 2 HD sources, and collaboration with optimal definition for the best video quality every time, regardless of environment.
- Highest Quality Audio with flexibility to add up to 2 microphones directly from the codec, and superior, full duplex audio with high quality stereo sound.
- Full APIs.
- Ensure successful, streamlined integration projects with standards-compliant professional connectors.
- The Codec C40 ties easily into the TANDBERG Total Solution for full management, firewall traversal and advanced services.

### Codec C40 and C60 Comparison

<table>
<thead>
<tr>
<th></th>
<th>Codec C40</th>
<th>Codec C60</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Resolution</strong></td>
<td>P2P: 1080p30/720p60</td>
<td>P2P: 1080p30 / 720p60</td>
</tr>
<tr>
<td><strong>Audio inputs</strong></td>
<td>Microphone: 2 x XLR</td>
<td>Microphone: 4 x XLR</td>
</tr>
<tr>
<td><strong>Video inputs</strong></td>
<td>2 x HDMI</td>
<td>2 x HDMI</td>
</tr>
<tr>
<td></td>
<td>1 x DVI</td>
<td>2 x DVI</td>
</tr>
<tr>
<td></td>
<td>1 x S-Video/Composite</td>
<td>1 x S-Video/Composite</td>
</tr>
<tr>
<td><strong>Video outputs</strong></td>
<td>1 x HDMI</td>
<td>1 x HDMI</td>
</tr>
<tr>
<td></td>
<td>1 x DVI</td>
<td>1 x DVI</td>
</tr>
<tr>
<td></td>
<td>1 x Composite</td>
<td>1 x Composite</td>
</tr>
<tr>
<td><strong>MultiSite</strong></td>
<td>Sites: 4 way CP</td>
<td>Sites: 4 way CP</td>
</tr>
<tr>
<td></td>
<td>Resolution: 576p30</td>
<td>Resolution: 720p30</td>
</tr>
<tr>
<td><strong>Other</strong></td>
<td>Application Programmer Interface (API)</td>
<td>Application Programmer Interface (API)</td>
</tr>
<tr>
<td></td>
<td>Video compositing API</td>
<td>Video compositing API</td>
</tr>
<tr>
<td></td>
<td>GPIO</td>
<td>GPIO</td>
</tr>
</tbody>
</table>
What’s new in this version

This section gives you an overview of the new and changed API commands for Codec C60 in the TC3.0 software version.

You will find detailed descriptions of all the new and changed settings in the xConfiguration, xCommand, and xStatus sections.

Software release notes

For a complete overview of the news and changes, we recommend reading the TANDBERG TC Software Release Notes (TC3).

Go to: http://www.tandberg.com/docs

User documentation

You may already be familiar with the user guides for the Profile Series, Codec C Series and Quick Set C20. To help you find the information you need, TANDBERG has split the guides into smaller and more accessible parts.

The Administrator Guide has been split into:

- Getting started guide
- Video conference room primer guide
- Camera user guide
- Administrator guide (smaller size)
- Legal and safety information

The System Integrator Guide has been split into:

- API guides for Codec C90, C60, C40
- Physical interfaces guides for Codec C90, C60, C40

The User Guides have been merged into one guide:

- User Guide for Profile Series, Codec C Series and Quick Set C20

The user documentation is available from our web site. Select your product from the drop down list to see an overview of the user documentation for that product.

Go to: http://www.tandberg.com/docs

New commands

xConfiguration

- Audio Input Microphone Equalizer Mode
- Audio Input Microphone Equalizer ID
- Audio Input Line Equalizer Mode
- Audio Input Line Equalizer ID
- Audio Output Line Equalizer Mode
- Audio Output Line Equalizer ID
- Audio Input HDMI Video Association Mute On Inactive Video
- Audio Input HDMI Video Association Video Input Source
- Audio Input Line Video Association Mute On Inactive Video
- Audio Input Line Video Association Video Input Source
- Audio Input Microphone Video Association Mute On Inactive Video
- Audio Input Microphone Video Association Video Input Source
- Cameras Power Line Frequency
- Cameras DHCP
- Conference Mic Unmute On Disconnect
- Conference Video Bandwidth Mode
- Conference Video Bandwidth Main Channel Weight
- Conference Video Bandwidth Presentation Channel Weight
- GPIO Pin Mode*
- Network Services HTTPS Verify Server Certificate
- Provisioning External Manager Protocol
- Video Input Source Optimal Definition Profile
- Video Input Source Optimal Definition Threshold 60fps
- Video Output HDMI Overscan Level
- Video Output DVI Overscan Level
- Video Output Composite Overscan Level*
- OSD Today’s Bookings
- Video OSD Input Method Input Language
- Video OSD Input Method Cyrillic

* Not supported on Codec C40
New commands, continued...

**xCommand**
- xCommand Audio Equalizer List
- xCommand Audio Equalizer Update
- xCommand GPIO Manual State Set*
- xCommand SystemUnit ConfigurationProfile Change
- xCommand SystemUnit ConfigurationProfile Remove
- xCommand SystemUnit ConfigurationProfile SaveCurrentConfigurationAs
- xCommand SystemUnit ConfigurationProfile List
- xCommand SystemUnit ConfigurationProfile CancelChange
- xCommand SystemUnit FactoryReset
- xCommand Video Layout AutoMode SetLayoutFamily
- xCommand Video Layout AutoMode Reset
- xCommand Video Layout AutoMode List
- xCommand Video Layout ListLayoutFamily
- xCommand Video Layout ListLayoutGraphic
- xCommand Message Alert Display
- xCommand Message Alert Clear
- xCommand Message Prompt Display
- xCommand Message Prompt Clear
- xCommand Message Prompt Response
- xCommand Message Alert Display

**xStatus**
- xStatus Camera IpAddress
- xStatus Camera MacAddress
- xStatus GPIO Pin State*
- xStatus SIP Profile Proxy Status
- xStatus SIP Profile Proxy Address
- xStatus SIP Profile Secure
- xStatus SIP Profile Verified
- xStatus SIP Profile Authentication
- xStatus SIP Profile Registration Status
- xStatus SIP Profile Registration Reason
- xStatus SIP Profile Registration URI
- xStatus SystemUnit ProductPlatform

Commands that have been changed

**xConfiguration**
- xConfiguration Camera Flip
  - The Auto parameter has been added.
- xConfiguration Camera Mirror
  - The Auto parameter has been added.
- xConfiguration SerialPort BaudRate
  - The 57600 parameter has been added.

**xCommand**
- xCommand Audio Sound Play
  - The Dial parameter has been added.
- xCommand Preset Store
  - The Description parameter is now optional.
- xCommand Phonebook Search
  - The SearchString parameter is now optional.
  - Added results set for timeout:
    ResultSet Error MessageType: Timeout.
- xCommand Phonebook Contact Add
  - The ImageURL parameter has been added.
  - The Title parameter has been added.
- xCommand Phonebook Contact Modify ImageURL
  - The ImageURL parameter has been added.
  - The Title parameter has been added.
- xCommand Phonebook Contact Modify Title
  - The Title parameter has been added.
- xCommand Video Layout Add
  - The LayoutId parameter has been added.
- xCommand Video Layout Frame Add
  - The LayoutId parameter has been added.

Commands that are removed

**xConfiguration**
- xConfiguration Video Encoder Threshold60fps
  - Removed.

**xCommand**
- xCommand Camera Upgrade
  - Removed.

* Not supported on Codec C40
Chapter 2

About the TANDBERG API
TANDBERG API

Basic Principles

The heart of the API is the TANDBERG 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 TANDBERG 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 hierarchical 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

Main types of information

If we look at the TANDBERG systems we can identify three 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, assigning floor to a specific site, disconnecting 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.

The TANDBERG API-Engine

The TANDBERG API-Engine is optimized for easy, yet advanced, machine-machine interaction between a TANDBERG 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 TANDBERG 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).
Addressing Using XPath or TANDBERG SimplePath

To address information in the hierarchic structure of Status and Configuration information, the TANDBERG systems support abbreviated XML Path Language (XPath) and a proprietary notation called TANDBERG 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 TANDBERG 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 TANDBERG 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 TANDBERG SimplePath when presenting configurations.

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

The structuring of information together with XPath and TANDBERG 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 TANDBERG 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
• Default password is: TANDBERG

Serial port login

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

• User name is: admin
• Default password is: TANDBERG

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: <9600/19200/38400/115200>

• 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 TANDBERG 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 TANDBERG 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 TANDBERG 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.
## 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 values types

<table>
<thead>
<tr>
<th>Value Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer values: <code>&lt;x..y&gt;</code></td>
<td>Defines the valid range for an integer input. x = min value, y = max value.</td>
</tr>
<tr>
<td><code>&lt;1..100&gt;</code></td>
<td></td>
</tr>
<tr>
<td>Literal values: <code>&lt;X/Y/.../Z&gt;</code></td>
<td>Defines the possible values for a given configuration.</td>
</tr>
<tr>
<td><code>&lt;On/Off/Auto&gt;</code></td>
<td></td>
</tr>
<tr>
<td>String values: <code>&lt;S: x, y&gt;</code></td>
<td>Defines that the valid input for this configuration is a String with minimum length x and maximum length of y characters.</td>
</tr>
<tr>
<td><code>&lt;S: 0, 49&gt;</code></td>
<td></td>
</tr>
<tr>
<td>IPv4 Address values: <code>&lt;IPAddr&gt;</code></td>
<td>Defines that the input must be an IPv4 address.</td>
</tr>
<tr>
<td><code>&lt;IPAddr&gt;</code></td>
<td></td>
</tr>
<tr>
<td>IPv6 Address values: <code>&lt;IPv6Addr: x, y&gt;</code></td>
<td>Defines that the input must be an IPv6 address with minimum length x and maximum length y.</td>
</tr>
<tr>
<td><code>&lt;IPv6Addr: 0, 43&gt;</code></td>
<td></td>
</tr>
<tr>
<td>IPv4 or IPv6 Address values: <code>&lt;IPv4v6Addr: x, y&gt;</code></td>
<td>Defines that the input must be an IPv4 or IPv6 address with minimum length x and maximum length y.</td>
</tr>
<tr>
<td><code>&lt;IPv4v6Addr: 0, 43&gt;</code></td>
<td></td>
</tr>
</tbody>
</table>

* Not supported in this version
Root commands

By typing ? or help after connecting to the TANDBERG Codec C–Series using RS-232/Telnet/SSH, the system will list all supported root commands.
Main type of commands

The XACLI is divided into three main types of commands, reflecting the information types supported by the TANDBERG 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 TANDBERG codec with multiple sessions, each session can define feedback individually.

More on this can be found in xfeedback.

- xFeedback

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 TANDBERG 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 hierarchic tree structure.

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

- xConfiguration ?
- xConfiguration help

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

- xConfiguration ??

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

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

Example: xConfiguration Audio Input HDMI 1 Mode: On

The type
The path
The value

xConfiguration ?

- User Configurations -

Audio                GPIO                Phonebook                Standby
Cameras              H323                Provisioning            SystemUnit
Conference           Network              SerialPort              Time
Experimental         NetworkServices      SIP                    Video

OK

xConfiguration ??

*h xConfiguration Audio Volume: <0..100>
Sets the volume level [0-100] on the loudspeaker output in steps of 0.5dB from -34.5dB to 15dB. 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 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

Configuration 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 a system setting you can use a help query. Enter the path followed by ? or help.

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

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

- xConfiguration <address expression> ?

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 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>
About xCommand

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**

Command 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>
  OnlyAudio: <False/True>
  OK
```

**Example with xCommand Write**

Dial a number with only the required parameter:

```
xCommand Dial Number: 558458
OK
*r DialResult (status=OK):
  CallId: 2
  ConferenceId: 1
  *r/end
  OK
```
### About xStatus commands

Status type commands returns information about the system and system processes. Status type commands are read by the user. All status information is structured in a hierarchy, making up a database constantly being updated by the system to reflect system and process changes.

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

- `xStatus ?`
- `xStatus help`

#### 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

### Table: Supported Status Commands

<table>
<thead>
<tr>
<th>Category</th>
<th>Subcategory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Call</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CallTransfer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Camera</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conference</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diagnostics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MediaChannels</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Experimental</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GPIO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H323</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTPFeedback</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MediaChannels</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MediaChannels</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Preset</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standby</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SystemUnit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Video</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**OK**
Query status information

Status 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 TANDBERG SimplePath).

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

Address status information with xStatus

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

xStatus <address expression>

Example 1: Query all ongoing Call information:

xstatus call
*s Call 8 Status: Connected
*s Call 8 Direction: Outgoing
*s Call 8 Protocol: "h323"
*s Call 8 RemoteNumber: "558458"
*s Call 8 CallbackNumber: "h323:alice.wonderland.office@tandberg.com"
*s Call 8 DisplayName: "alice.wonderland.office@tandberg.com"
*s Call 8 CallRate: 768
*s Call 8 Encryption Type: "Aes-128"

OK

Example 2: Query the protocol for a call:

xstatus call protocol
*s Call 8 Protocol: "h323"

OK
About xHistory command

History 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: 2
*h xHistory CallLogs Call 1 Protocol: "h323"
*h xHistory CallLogs Call 1 RemoteNumber: "h323:john.doe@company.com"
*h xHistory CallLogs Call 1 CallbackNumber: "h323:john.doe@company.com"
*h xHistory CallLogs Call 1 DisplayName: "John Doe - Office"
*h xHistory CallLogs Call 1 CallRate: 4000
*h xHistory CallLogs Call 1 DisconnectCauseValue: 1
*h xHistory CallLogs Call 1 DisconnectCause: "MC:Normal"
*h xHistory CallLogs Call 1 StartTime: "2010/01/22 13:22:13"
*h xHistory CallLogs Call 1 Duration: 187
*h xHistory CallLogs Call 1 Encryption: "Aes-128"
...**
*h xHistory CallLogs Recent 2 CounterMissed: 0
*h xHistory CallLogs Recent 3 CounterMissed: 0
...**
*h xHistory CallLogs Outgoing 1 Counter: 1
*h xHistory CallLogs Outgoing 2 Counter: 1
...**
ex
```
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 TANDBERG codec with multiple sessions, each session can define feedback individually.

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

```
xFeedback ?

xFeedback help:
  xFeedback Register XPathExpression - 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
```

Example with xFeedback

```
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 TANDBERG 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.

```plaintext
xPreferences ?
xpreferences usage:
xpreferences outputmode <terminal/xml>
OK
```
TANDBERG XML API service
TXAS is a service provided by TANDBERG 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 which will cause a failure).

putxml
REQUEST:
   /putxml
PARAM:
   HTTP BODY as argument
Chapter 3

Description of the xConfiguration commands
Description of the xConfiguration commands
In the following pages you will find a complete list of the xConfiguration commands. The examples shows either the default value or an example of a value.

The Audio settings

xConfiguration Audio Input HDMI [2] Level
Defines the input level of HDMI input 2 in steps of 1dB from -24dB to 0dB.
See the Audio Level tables in the physical interfaces guide for the codec for a complete overview of the menu values represented in dB.
Valuespace: <-24..0>
Range: -24 to 0 dB
Example: xConfiguration Audio Input HDMI 3 Level: 0

Determines whether or not the two audio channels on the HDMI 2 input should be enabled.
Valuespace: <On/Off>
On: Set to On to enable the audio channels on the HDMI 2 input.
Off: Set to Off to disable the audio channels on the HDMI 2 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.
Valuespace: <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.
Valuespace: <1/2/3>
Range: Select one of the three video input sources for the HDMI input.
Example: xConfiguration Audio Input HDMI 2 VideoAssociation VideoInputSource: 1

xConfiguration Audio Input Line [1..2] Equalizer ID
Select equalizer ID[1...8].
Valuespace: <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
Determines whether or not the selected equalizer is enabled.
Valuespace: <On/Off>
On: Use the selected equalizer
Off: No equalizer
Example: xConfiguration Audio Input Line 1 Equalizer Mode: Off

xConfiguration Audio Input Line [1..2] VideoAssociation MuteOnInactiveVideo
Enable association of a video source to a Line audio input.
Valuespace: <On/Off>
On: A video source is associated, and the audio will be muted if the associated video source is not displayed.
Off: No video source is associated.
Example: xConfiguration Audio Input Line 1 VideoAssociation MuteOnInactiveVideo: Off

xConfiguration Audio Input Line [1..2] VideoAssociation VideoInputSource
Select the associated video input source.
Valuespace: <1/2/3>
Range: Select one of the three video input sources for the Line input.
Example: xConfiguration Audio Input Line 1 VideoAssociation VideoInputSource: 1

xConfiguration Audio Input Line [1..2] Channel
Defines whether the Audio Line input is a mono signal or part of a multichannel signal.
Valuespace: <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
The Audio settings, continued...

**xConfiguration Audio Input Line [1..2] Level**
Defines the input level of the selected Line input connector in steps of 1dB from 0dB to 24dB.

Valuespace: \(<0..24>\)

Range: 0 to 24 dB

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

**xConfiguration Audio Input Line [1..2] LoopSuppression**
Loop suppression detects whether a delayed signal loop is present from an audio Line output to an audio Line input on the codec. If a loop is detected, this unwanted feedback is suppressed.

Valuespace: \(<\text{Off}>\)

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

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

**xConfiguration Audio Input Line [1..2] Mode**
Determines whether or not the selected Audio Line input connector is enabled.

Valuespace: \(<\text{On/Off}>\)

On: Set to On to enable the Audio Line input.
Off: Set to Off to disable the Audio Line input.

Example: `xCConfiguration Audio Input Line 1 Mode: 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 equalizer ID[1..8]

Valuespace: \(<1..8>\)

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

**xConfiguration Audio Input Microphone [1..2]/[1..4] Equalizer Mode**
Determines whether or not the selected equalizer is enabled.

Valuespace: \(<\text{On/Off}>\)

On: Use the selected equalizer
Off: No equalizer

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

**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. Requires the Echo Control Mode to be enabled for the selected microphone.

Valuespace: \(<\text{On/Off}>\)

On: The Noise Reduction should be set to On in the presence of low frequency noise.
Off: Turns Noise Reduction Off for the selected microphone connector.

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

**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 connector.

Valuespace: \(<\text{On/Off}>\)

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: `xCConfiguration Audio Input Microphone 1 VideoAssociation MuteOnInactiveVideo: On`
The Audio settings, continued...

**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.

Valuespace:  <1/2/3>
Range: Select one of the five video input sources for the microphone connector.

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.
Defines the input level of the selected microphone connector in steps of 1dB from -24dB to 0dB.
See the Audio Level tables in the physical interfaces guide for the codec for a complete overview of the values represented in dB.

Valuespace:  <-24..0>
Range: -24 to 0dB

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.
Determines whether or not the selected microphone connector is enabled.

Valuespace:  <On/Off>
On: Set to On to enable the microphone connector.
Off: Set to Off to 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 selected microphone connector can be set to line or microphone mode.

Valuespace:  <Microphone/Line/>
Microphone: 48 V 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.

Example: xConfiguration Audio Input Microphone 1 Type: Line

---

The Audio settings, continued...

**xConfiguration Audio Output HDMI [1..1] Level**
Defines the output level of HDMI output 1 in steps of 1dB from -24dB to 0dB.
See the Audio Level tables in the physical interfaces guide for the codec for a complete overview of the values represented in dB.

Valuespace:  <-24..0>
Range: -24 to 0dB

Example: xConfiguration Audio Output HDMI 1 Level: 0

**xConfiguration Audio Output HDMI [1..1] Mode**
Determines whether or not the audio channel on the HDMI output should be enabled.

Valuespace:  <On/Off>
On: Set to On to enable the audio channel on the HDMI 1 or HDMI 3 output.
Off: Set to Off to disable the audio channel on the HDMI 1 or HDMI 3 output.

Example: xConfiguration Audio Output HDMI 1 Mode: On

**xConfiguration Audio Output Line [1..2] Equalizer ID**
Select equalizer ID[1...8]

Valuespace:  <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**
Determines whether or not the selected equalizer is enabled.

Valuespace:  <On/Off>
On: Use the selected equalizer
Off: No equalizer

Example: xConfiguration Audio Output Line 1 Equalizer Mode: Off

**xConfiguration Audio Output Line [1..2] Channel**
Defines whether the Audio Line output is a mono signal or part of a multichannel signal.

Valuespace:  <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
The Audio settings, continued...

**xConfiguration Audio Output Line [1..2] Level**
Defines the output level of the selected Audio Output Line connector in steps of 1dB from -24dB to 0dB. See the Audio Level tables in the physical interfaces guide for the codec for a complete overview of the values represented in dB.

Valuespace: <-24..0>

Range: -24 to 0dB

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

**xConfiguration Audio Output Line [1..2] Mode**
Determines whether or not the selected Audio Line output connector is enabled.

Valuespace: <On/Off>

- **On**: Set to On to enable the Audio Line output.
- **Off**: Set to Off to disable the Audio Line output.

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

**xConfiguration Audio Output Line [1] Type**
Determines if the Audio Line output is an analog or digital type output.

Valuespace: <Auto/SPDIF>

- **Auto**: If a TANDBERG Digital NAM is detected then SPDIF mode will be selected, otherwise analog mode will be selected.
- **SPDIF**: Set to SPDIF when you want the Audio Line 1 output to be in digital mode.

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

**xConfiguration Audio Output Line [2] Type**
Applies to Line 2: This is an analog output.

Valuespace: <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.

Valuespace: <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.

Valuespace: <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 ringtone volume[0-100] for an incoming call in steps of 0.5dB from -34.5dB to 15dB.

Valuespace: <0..100>

Range: Select a value from 0 to 100. Volume 0 = Off.

Example: `xConfiguration Audio SoundsAndAlerts RingVolume: 50`

**xConfiguration Audio Volume**
Sets the volume level [0-100] on the loudspeaker output in steps of 0.5dB from -34.5dB to 15dB.

Valuespace: <0..100>

Range: Select a value from 0 to 100. Volume 0 = Off.

Example: `xConfiguration Audio Volume: 70`
The Camera settings

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

Valuespace: `<Auto/50Hz/60Hz>`
- Auto: Set to Auto to enable power frequency auto detection in the camera.
- 50Hz: Set to 50Hz.
- 60Hz: Set to 50Hz.

**Example:** xConfiguration Cameras PowerLine Frequency: Auto

**xConfiguration Cameras Camera [1..7] Flip**
With Flip mode (vertical flip) you can flip the image upside down.

Valuespace: `<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**
Determines whether the selected camera should be in auto focus or manual focus mode.

Valuespace: `<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: If set to Manual the focus is adjusted manually.

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

**xConfiguration Cameras Camera [1..7] Gamma Level**
By setting the Gamma Level you can select which gamma correction table to use. This setting may be useful in difficult lighting conditions, where changes to the brightness setting does not provide satisfactory results. Requires the Gamma Mode to be set to manual.

Valuespace: `<0..7>`
- Range: Select a value from 0 to 7.

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

**xConfiguration Cameras Camera [1..7] Gamma Mode**
By setting the Gamma Level you can select which gamma correction table to use. This setting may be useful in difficult lighting conditions, where changes to the brightness setting does not provide satisfactory results. Requires the Gamma Mode to be set to Manual.

Valuespace: `<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: If set to Manual the focus is adjusted manually.

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

**xConfiguration Cameras Camera [1..7] Brightness Level**
Define the Brightness Level for the selected camera. Requires the Brightness Mode to be set to manual.

Valuespace: `<1..31>`
- Range: Select a value from 1 to 31.

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

**xConfiguration Cameras Camera [1..7] Brightness Mode**
Define whether to control the selected camera brightness manually or to have it automatically adjusted by the system.

Valuespace: `<Auto/Manual>`
- Auto: When set to Auto, the camera brightness is automatically set by the system.
- Manual: Set to Manual to 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] Backlight**
Backlight is used to compensate for lights shining directly at the camera (usually the sun entering the window) to avoid a too dark image from the room.

Valuespace: `<On/Off>`
- On: Set to On to turn on the backlight compensation.
- Off: Set to Off to turn the backlight compensation off.

**Example:** xConfiguration Cameras Camera 1 Backlight: Off

---

*The Camera settings, continued...*
The Camera settings, continued...

**xConfiguration Cameras Camera [1..7] IrSensor**
The Camera IR setting determines whether the infrared receiver at the camera should be enabled or not. The IR sensor LED is located in the front of the camera and flickers when the IR sensor is activated from the remote control.

Valuespace:  <On/Off>
  - On: Set to On to enable the IR sensor on the camera.
  - Off: Set to Off to 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.

Valuespace:  <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: Set to On to 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: Set to Off to 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 Level**
Specify which camera to control. Define the Whitebalance Level for the selected camera. Requires the Whitebalance Mode to be set to manual.

Valuespace:  <1..16>
  - Range: Select a value from 1 to 16.

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

**xConfiguration Cameras Camera [1..7] Whitebalance Mode**
Define whether to control the selected camera whitebalance manually or to have it automatically adjusted by the system.

Valuespace:  <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
The Conference settings

**xConfiguration Conference [1..1] AutoAnswer Delay**
Defines how long (in seconds) an incoming call has to wait before it is answered automatically by the system. Requires the Autoanswer Mode to be enabled.

Valuespace: `<0..50>`
Range: 0-50 seconds

*Example:* `xConfiguration Conference 1 AutoAnswer Delay: 0`

**xConfiguration Conference [1..1] AutoAnswer Mode**
The Autoanswer setting determines whether an incoming call is put through automatically or manually.

Valuespace: `<On/Off>`
- **On:** The system will automatically answer all incoming calls.
- **Off:** All incoming call 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**
The Autoanswer Mute setting determines whether the microphone is muted when an incoming call is automatically answered.

Valuespace: `<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] MicUnmuteOnDisconnect**
The MicUnmuteOnDisconnect setting determines if the microphones should be automatically unmuted 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.

Valuespace: `<On/Off>`
- **On:** Microphones will be unmuted when all calls are disconnected
- **Off:** Microphones will not be unmuted when all calls are disconnected

*Example:* `xConfiguration Conference 1 MicUnmuteOnDisconnect: On`

**xConfiguration Conference [1..1] DoNotDisturb Mode**
The Do Not Disturb setting determines whether or not there should be an alert on incoming calls.

Valuespace: `<On/Off>`
- **On:** Set to On when you want no alert to incoming calls. The calling side will receive a busy signal when trying to call the codec.
- **Off:** This is the default setting. The DoNotDisturb is automatically turned Off if the codec receives any IR signal from the handheld remote control.

*Example:* `xConfiguration DoNotDisturb Mode: Off`

**xConfiguration Conference [1..1] IncomingMultisiteCall Mode**
The Incoming Multisite Call setting determines whether or not the system should accept incoming calls to an already active conference.

Valuespace: `<Allow/Deny>`
- **Allow:** When set to Allow, and with an ongoing MCU call/conference, the user can accept another incoming call. This will result in the incoming call being 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] 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).

Valuespace: `<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] Encryption Mode
In Point to point calls (BestEffort mode): 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 (BestEffort mode): In order to have encrypted MultiSite conferences, all sites must support encryption. If not, the conference will be unencrypted.

Icons on screen: A padlock with the text “Encryption On” displays on screen, for a few seconds, when the conference starts.

Valuespace:  <Off/On/BestEffort>

- **BestEffort**: The system will use encryption whenever possible.
- **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
Specify the Default Call Protocol to be used when placing calls from the system. The call protocol can also be defined directly for each call when setting up a call.

Valuespace:  <H323/SIP>

- **H.323**: Select H.323 to ensure that calls are set up as a H.323 calls.
- **SIP**: Select SIP to ensure that calls are set up as a SIP calls.

Example: xConfiguration Conference 1 DefaultCall Protocol: H323

xConfiguration Conference [1..1] DefaultCall Rate
Specify the Default Call Rate to be used when placing calls from the system. The call rate can also be defined directly for each call when setting up a call.

Valuespace:  <64..6000>

- **Range**: 64-6000kbps

Example: xConfiguration Conference 1 DefaultCall Rate: 768

xConfiguration Conference [1..1] VideoBandwidth Mode
In “Dynamic” mode the available transmit bandwidth for the video channels is distributed among the currently active channels. This means that the main video channels will use the bandwidth of the presentation channel, if there is no active presentation. In “Static” mode the bandwidth is assigned to each video channel even if it is not active.

Valuespace:  <Dynamic/Static>

- **Dynamic**: The available transmit bandwidth for the video channels are distributed among the currently active channels.
- **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.

Valuespace:  <1..10>

- **Range**: Select a value from 0 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.

Valuespace:  <1..10>

- **Range**: Select a value from 0 to 10.

Example: xConfiguration Conference 1 VideoBandwidth PresentationChannel Weight: 5
The GPIO settings

**xConfiguration GPIO Pin [1..4] Mode**

NOTE: This command is not supported on Codec C40.

The four GPIO pins can be 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.

Valuespace: <InputNoAction/OutputManualState/OutputInCall/OutputMicrophonesMuted/
OutputPresentationOn/OutputAllCallsEncrypted/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.
- **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 Profile [1..1] Authentication LoginName**

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

Valuespace: <S: 0, 50>

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

**xConfiguration H323 Profile [1..1] Authentication Password**

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

Valuespace: <S: 0, 50>

**Example**: xConfiguration H323 Profile 1 Authentication Password:

**xConfiguration H323 Profile [1..1] Authentication Mode**

Set authentication mode.

Valuespace: <On/Off>

- **On**: If the H.323 Gatekeeper Authentication Mode is set to On and a H.323 Gatekeeper indicates that it requires authentication, the system will try to authenticate itself to the gatekeeper. Requires the Authentication ID and Authentication Password to be defined on both the codec and the Gatekeeper.
- **Off**: If the H.323 Gatekeeper Authentication Mode is set to Off the system will not try to authenticate itself to a H.323 Gatekeeper, but will still try a normal registration.

**Example**: xConfiguration H323 Profile 1 Authentication Mode: Off
The H.323 settings, continued...

**xConfiguration H323 Profile [1..1] CallSetup Mode**

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

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

**Valuespace:** `<Direct/Gatekeeper>`

- **Direct:** An IP-address must be used when dialling in order to make the H.323 call.
- **Gatekeeper:** The system will use a Gatekeeper to make a H.323 call. When selecting this option the H.323 Profile Gatekeeper Address and H.323 Profile Gatekeeper Discovery settings must also be configured.

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

**xConfiguration H323 Profile [1..1] Gatekeeper Address**

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

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

- **Format:** String with a maximum of 255 characters.

**Example:** `xConfiguration H323 Profile 1 Gatekeeper Address: "10.47.1.58"`

**xConfiguration H323 Profile [1..1] Gatekeeper Discovery**

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

**Valuespace:** `<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] 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.

**Valuespace:** `<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"`

The H.323 settings, continued...

**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”

**Valuespace:** `<S: 0, 49>`

- **Format:** String with a maximum of 49 characters

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

**xConfiguration H323 Profile [1..1] PortAllocation**

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

**Valuespace:** `<Dynamic/Static>`

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

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

**Example:** `xConfiguration H323 Profile 1 PortAllocation: Dynamic`
The Network settings

xConfiguration Network [1..1] Assignment
Defines whether to use DHCP or Static IP assignment.
Changes to this setting requires a restart of the codec.
Valuespace: <Static/DHCP>
  Static: The IP Address, Subnet Mask and Default Gateway for the system must be specified in the respective address fields.
  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".
Valuespace: <S: 0, 64>
  Format: String with a maximum of 64 characters.
Example: xConfiguration Network 1 DNS Domain Name: "company.com"

xConfiguration Network [1..1] DNS Server [1..5] Address
Defines 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.
Valuespace: <S: 0, 64>
  Format: String with a maximum of 64 characters.
Example: xConfiguration Network 1 DNS Server 1 Address: ""

xConfiguration Network [1..1] IEEE8021X AnonymousIdentity
The 802.1X Anonymous ID string is to be used as unencrypted identity with EAP types that support different tunneled identity, like EAP-PEAP and EAP-TTLS. If set, the anonymous ID will be used for the initial (unencrypted) EAP Identity Request.
Valuespace: <S: 0, 64>
  Format: String with a maximum of 64 characters.
Example: xConfiguration Network 1 IEEE8021X AnonymousIdentity: ""

The Network settings, continued...

xConfiguration Network [1..1] IEEE8021X Eap Md5
Message-Digest algorithm 5. Is a Challenge Handshake Authentication Protocol that relies on a shared secret. MD5 is a Weak security. EAP - Extensible Authentication Protocol. MD5 - Message Digest Algorithm 5.
Valuespace: <On/Off>
  On: The EAP-MD5 protocol is enabled. Default mode is On.
  Off: The EAP-MD5 protocol is disabled.
Example: xConfiguration Network 1 IEEE8021X Eap Md5: On

xConfiguration Network [1..1] IEEE8021X Eap Peap
Valuespace: <On/Off>
  On: The EAP-PEAP protocol is enabled. Default mode is On.
  Off: The EAP-PEAP protocol is disabled.
Example: xConfiguration Network 1 IEEE8021X Eap Peap: On

xConfiguration Network [1..1] IEEE8021X Eap Ttls
Valuespace: <On/Off>
  On: The EAP-TTLS protocol is enabled. Default mode is On.
  Off: The EAP-TTLS protocol is disabled.
Example: xConfiguration Network 1 IEEE8021X Eap Ttls: On

xConfiguration Network [1..1] IEEE8021X Identity
The 802.1X Identity is the user name needed for 802.1X authentication.
Valuespace: <S: 0, 64>
  Format: String with a maximum of 64 characters.
Example: xConfiguration Network 1 IEEE8021X Identity: ""
The Network settings, *continued*...

**xConfiguration Network [1..1] IEEE8021X Mode**
The system may 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.

Valuespace: <On/Off>
- **On**: The 802.1X authentication is enabled.
- **Off**: The 802.1X authentication is disabled. Default mode is Off.

*Example:* `xConfiguration Network 1 IEEE8021X Mode: Off`

**xConfiguration Network [1..1] IEEE8021X Password**
The 802.1X Password is the password needed for 802.1X authentication.

Valuespace: <S: 0, 32>
- **Format**: String with a maximum of 32 characters.

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

**xConfiguration Network [1..1] IPv4 Address**
Defines the Static IP address for the system. Only applicable if Static IP assignment is chosen.

Valuespace: <S: 0, 64>
- **Format**: Compact string with a maximum of 64 characters.

*Example:* `xConfiguration Network 1 IPv4 Address: "10.47.5.100"`

**xConfiguration Network [1..1] IPv4 Gateway**
Defines the IP default gateway. Only applicable if Static IP assignment is chosen.

Valuespace: <S: 0, 64>
- **Format**: Compact string with a maximum of 64 characters.

*Example:* `xConfiguration Network 1 IPv4 Gateway: "10.47.5.100"`

**xConfiguration Network [1..1] IPv4 SubnetMask**
Defines the IP subnet mask. Only applicable if Static IP assignment is chosen.

Valuespace: <S: 0, 64>
- **Format**: Compact string with a maximum of 64 characters.

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

The Network settings, *continued*...

**xConfiguration Network [1..1] IPv4 QoS Mode**
Defines whether IP Diffserv QoS should be used. 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.

Valuespace: <Off/Diffserv>
- **Off**: When set to Off no QoS method is used.
- **Diffserv**: Select Diffserv and then go to the Diffserv sub-menus (Audio, Data, Signalling and Video) to configure these settings.

*Example:* `xConfiguration Network 1 IPv4 QoS Mode: diffserv`

**xConfiguration Network [1..1] IPv4 QoS Diffserv Audio**
The DiffServ Audio setting is used to define 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.

Valuespace: <0..63>
- **Audio**: A recommended value is DiffServ Code Point (DSCP) is AF41, which equals the value 34. If in doubt, contact your network administrator.

*Example:* `xConfiguration Network 1 IPv4 QoS Diffserv Audio: 0`

**xConfiguration Network [1..1] IPv4 QoS Diffserv Data**
The DiffServ Data setting is used to define 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.

Valuespace: <0..63>
- **Data**: A recommended value is DiffServ Code Point (DSCP) AF23, which equals the value 22. If in doubt, contact your network administrator.

*Example:* `xConfiguration Network 1 IPv4 QoS Diffserv Data: 0`
The Network settings, continued...

**xConfiguration Network [1..1] IPv4 QoS Diffserv Signalling**

The DiffServ Signalling setting is used to define 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.

Valuespace: <0..63>

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

Example: xConfiguration Network 1 IPv4 QoS Diffserv Signalling: 0

---

**xConfiguration Network [1..1] IPv4 QoS Diffserv Video**

The DiffServ Video setting is used to define 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.

Valuespace: <0..63>

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

Example: xConfiguration Network 1 IPv4 QoS Diffserv Video: 0

---

**xConfiguration Network [1..1] MTU**

Set the ethernet MTU (Maximum Transmission Unit).

Valuespace: <400..1500>

*Range:* Select a value from 400 to 1500.

Example: xConfiguration Network 1 MTU: 1500

---

**xConfiguration Network [1..1] Speed**

Set the ethernet link speed.

Valuespace: <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**

Configure how video packets transmission speed shall be controlled.

Valuespace: <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
The NetworkServices settings

xConfiguration NetworkServices H323 Mode
Determines whether the system should be able to place and receive H.323 calls.
NOTE! Changes in this setting requires the codec to be restarted.
Valuespace: <On/Off>
  On: Set to On to enable the possibility to place and receive H.323 calls. This is the default setting.
  Off: Set to Off to disable the possibility to place and receive H.323 calls.
Example: xConfiguration NetworkServices H323 Mode: On

xConfiguration NetworkServices HTTP Mode
HTTP is a web-interface for system management, call management such as call transfer, diagnostics and software uploads.
Valuespace: <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.
Valuespace: <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. This setting tells the system if it should verify that the certificate is signed by a trusted Certificate Authority (CA). This requires that list of trusted CAs is uploaded to the system in advance.
Valuespace: <On/Off>
  On: Verify server certificates.
  Off: Do not verify server certificates.
Example: xConfiguration NetworkServices HTTPS VerifyServerCertificate: Off

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.
Valuespace: <S: 0, 64>
  Format: String with a maximum of 64 characters.
Example: xConfiguration NetworkServices NTP Address: “1.tandberg.pool.ntp.org”

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.
Valuespace: <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 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 TANDBERG Management Suite (TMS) you must make sure the same SNMP Community is configured there too. Note! The SNMP Community password is case sensitive.
Valuespace: <S: 0, 50>
  Format: String with a maximum of 50 characters.
Example: xConfiguration NetworkServices SNMP CommunityName: “public”
The NetworkServices settings, continued...

xConfiguration 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 dialing, system disconnecting, MCU call, packet loss etc. Traps can be sent to multiple SNMP Trap Hosts.
Valuespace:  <S: 0, 64>
Format: String with a maximum of 64 characters.
Example: xConfiguration NetworkServices SNMP Host 1 Address: ""

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.
Valuespace:  <Off/ReadOnly/ReadWrite>
Off: Set to Off when you want to disable the SNMP network service.
ReadOnly: Set to ReadOnly when you want to enable the SNMP network service for queries only.
ReadWrite: Set to ReadWrite when you want to enable the SNMP network service for both queries and commands.
Example: xConfiguration NetworkServices SNMP Mode: ReadWrite

xConfiguration NetworkServices SNMP SystemContact
Enter the name of the Network Services SNMP System Contact.
Valuespace:  <S: 0, 50>
Format: String with a maximum of 50 characters.
Example: xConfiguration NetworkServices SNMP SystemContact: ""

xConfiguration NetworkServices SNMP SystemLocation
Enter the name of the Network Services SNMP System Location.
Valuespace:  <S: 0, 50>
Format: String with a maximum of 50 characters.
Example: xConfiguration NetworkServices SNMP SystemLocation: ""

The NetworkServices settings, continued...

xConfiguration NetworkServices Telnet Mode
Telnet is a network protocol used on the Internet or local area network (LAN) connections.
Valuespace:  <On/Off>
On: The Telnet protocol is enabled.
Off: The Telnet protocol is disabled. This is the default factory setting.
Example: xConfiguration NetworkServices Telnet Mode: Off

The Phonebook settings

xConfiguration Phonebook Server [1..5] ID
Enter a name for the external phonebook.
Valuespace:  <S: 0, 64>
Format: String with a maximum of 64 characters.
Example: xConfiguration Phonebook Server 1 ID: ""

xConfiguration Phonebook Server [1..5] URL
Enter the address (URL) to the external phonebook server.
Valuespace:  <S: 0, 255>
Format: String with a maximum of 255 characters.
The Provisioning settings

xConfiguration Provisioning ExternalManager Address
Specifies 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.

Valuespace: <S: 0, 64>
Format: String with a maximum of 64 characters.
Example: xConfiguration Provisioning ExternalManager Address: ""

xConfiguration Provisioning ExternalManager Path
Specifies 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.

Valuespace: <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
Determines whether or not to use secure management.

Valuespace: <HTTP/HTTPS>
- HTTP: Set to HTTP to disable secure management. Requires HTTP to be enabled in the Network Services HTTP Mode setting.
- HTTPS: Set to HTTPS to enable secure management. Requires HTTPS to be enabled in the Network Services HTTPS Mode setting.

Example: xConfiguration Provisioning ExternalManager Protocol: http

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

Valuespace: <Off/TMS>
- Off: The system will not try to register to any management system.
- TMS: If set to TMS the system will try to register with a TMS server as described in Provisioning ExternalManager settings. TMS is short for TANDBERG Management System. Contact your TANDBERG representative for more information.

Example: xConfiguration Provisioning Mode: TMS

The SerialPort settings

xConfiguration SerialPort BaudRate
Specify the baud rate on the COM port (data port). The default value is 38400.

Other default parameters for the COM port are: Parity: None Databits: 8 Stopbits: 1 Flow control: None.

Valuespace: <9600/19200/38400/57600/115200>
Range: Select a baud rate from the list of baud rates (bps).

Example: xConfiguration SerialPort BaudRate: 38400

xConfiguration SerialPort LoginRequired
The Serial Login setting determines whether or not there should be a login when connecting to the COM port (data port).

Valuespace: <On/Off>
- On: Login is required when connecting to the COM port (data port).
- Off: The user can access the COM port (data 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.
Valuespace: <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.
Valuespace: <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.
Valuespace: <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] Outbound**
Valuespace: <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  

The SIP settings, continued...

**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.
Valuespace: <S: 0, 255>
Format: Compact string with a maximum of 255 characters.
**Example:** xConfiguration SIP Profile 1 Proxy 1 Address: ""  

**xConfiguration SIP Profile [1..1] Proxy [1..4] Discovery**
Valuespace: <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
Valuespace: <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 a Alcatel-Lucent OmniPCX Enterprise R7 or later.
- **Avaya:** Must be used when registered to a 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 auto is 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.
Valuespace: <S: 0, 255>
Format: Compact string with a maximum of 255 characters.
**Example:** xConfiguration SIP Profile 1 URI: “sip:name@example.com”
The Standby settings

**xConfiguration Standby BootAction**
Decide what the system is going to do on boot.

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

None: No action.

Preset 1..15: Activate the selected preset.

RestoreCameraPosition: Set the camera to the position it had before the last boot.

DefaultCameraPosition: Set the camera to the factory default position.

Example: `xConfiguration Standby BootAction: DefaultCameraPosition`

**xConfiguration Standby Control**
Determine whether the system should go into standby mode or not.

Valuespace: `<On/Off>`

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

Off: Not entering 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. Requires the Standby Control to be enabled.

Valuespace: `<1..480>`

Range: 1-480 minutes

Example: `xConfiguration Standby Delay: 10`

**xConfiguration Standby StandbyAction**
NEW TC2.1 Decide what the system is going to do when going into standby mode.

Valuespace: `<None/PrivacyPosition>`

None: No action.

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

Example: `xConfiguration Standby StandbyAction: PrivacyPosition`

**The Standby settings, continued...**

**xConfiguration Standby WakeupAction**
Decide what the system is going to do when leaving standby mode.

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

None: No action.

Preset 1..15: Activate the selected preset.

RestoreCameraPosition: Set the camera to the position it had before entering standby.

DefaultCameraPosition: Set the camera to the factory default position.

Example: `xConfiguration Standby WakeupAction: RestoreCameraPosition`
The SystemUnit settings

**xConfiguration SystemUnit CallLogging Mode**
Enables/disables logging of calls received or placed by the system. The call logs may then be viewed via the GUI or using the xHistory command.

Valuespace: <On/Off>
- On: Enable logging.
- Off: Disable logging.

Example: xConfiguration SystemUnit CallLogging Mode: On

**xConfiguration SystemUnit IrSensor Mode**
The System Unit IR Sensor setting determines whether the infrared receiver on the codec should be enabled or not. The IR sensor LED is located in the front of the codec and flickers when an IR signal is received from the remote control.

Valuespace: <On/Off/Auto>
- On: Set to On to enable the IR sensor on the codec.
- Off: Set to Off to 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: On

**xConfiguration SystemUnit MenuLanguage**
The setting is used to select the language for the GUI (Graphical User Interface).

Valuespace: <English/Norwegian/Swedish/German/French/Italian/Japanese/Russian/Spanish/Korean/Finnish/ChineseSimplified/ChineseTraditional/PortugueseBrazilian/Turkish/Polish>

Example: xConfiguration SystemUnit MenuLanguage: English

**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:

* When the codec is acting as an SNMP Agent
* Towards a DHCP server

Valuespace: <S: 0, 50>

Format: String with a maximum of 50 characters.

Example: xConfiguration SystemUnit Name: "Meeting Room Name"

**xConfiguration SystemUnit Time DateFormat**
Specifies the date format.

Valuespace: <DD_MM_YY/MM_DD_YY/YY_MM_DD>

DD_MM_YY: The date January 30th 2009 will be displayed: 30.01.09
MM_DD_YY: The date January 30th 2009 will be displayed: 01.30.09
YY_MM_DD: The date January 30th 2009 will be displayed: 09.01.30

Example: xConfiguration SystemUnit Time DateFormat: DD_MM_YY

**xConfiguration SystemUnit Time TimeFormat**
Specifies the time format.

Valuespace: <24H/12H>

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

Example: xConfiguration SystemUnit Time TimeFormat: 24H

The Time settings

**xConfiguration Time Date Format**
Specifies the date format.

Valuespace: <DD_MM_YY/MM_DD_YY/YY_MM_DD>

Example: xConfiguration Date Format: DD_MM_YY

**xConfiguration Time Time Format**
Specifies the time format.

Valuespace: <24H/12H>

Example: xConfiguration Time Time Format: 24H
The Time settings, continued...

**xConfiguration Time Zone**

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

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

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.

Valuespace: `<1..3>`

Range: Select on of the three presentation sources

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, this detection might fail. This setting makes it possible to override the AutoDetect and select the correct DVI video input.

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, this detection might fail. 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 Analog YPbPr setting has to be selected.

Valuespace: `<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 I 2 Type: AutoDetect

**xConfiguration Video Input Source [1..3] CameraControl Camerad**

Select the ID of the camera in the Visca chain that is connected to this camera source. The Camerad setting represents the camera's position in the Visca chain.

Cascaded cameras and Visca commands are described in the PrecisionHD 1080p User Guide. The user guide is found at www.tandberg.com/docs

Valuespace: `<1..5>`

Example: xConfiguration Video Input Source 1 CameraControl CameraId: 1
xConfiguration Video Input Source [1..3] CameraControl Mode
Determine whether or not the camera control should be enabled for the selected video input source.

Valuespace: <On/Off>
- On: Set to On to enable camera control for the camera connected to the selected video input connector.
- Off: Set to Off to disable camera control for the camera connected to the selected video input connector.

Example: xConfiguration Video Input Source 1 CameraControl Mode: On

xConfiguration Video Input Source 1 Connector
Select which video input connector to be active on connector group 1

Valuespace: <HDMI>
- HDMI: Select HDMI when you want to use the HDMI 1 connector as input

Example: xConfiguration Video Input Source 1 Connector: HDMI

xConfiguration Video Input Source 2 Connector
Select which video input connector to be active on connector group 2

Valuespace: <HDMI/DVI>
- HDMI: Select HDMI when you want to use the HDMI 2 connector as input
- DVI: Select DVI-I when you want to use the DVI-I-3 connector as input

Example: xConfiguration Video Input Source 2 Connector: HDMI

xConfiguration Video Input Source 3 Connector
Select which video input connector to be active on connector group 3

Valuespace: <DVI/Composite/YC>
- DVI: Select DVI-I when you want to use the DVI-I-3 connector as input.
- YC: Select YC when you want to use the S-Video (YC) input. Connect the S-Video input to the connector marked as Y/Comp and C.
- Composite: Select Comp when you want to use the Composite input. Connect the Composite input to the connector marked

Example: xConfiguration Video Input Source 3 Connector: DVI

The Video settings, continued...

xConfiguration Video Input Source [1..3] Name
Customizable name of the connector group. Enter the name of the video input source 1-3.

Valuespace: <S: 0, 50>
- Format: String with a maximum of 50 characters.

Example: xConfiguration Video Input Source 1 Name: ""

xConfiguration Video Input Source [1..3] OptimalDefinition Profile
The system will automatically choose what resolution and framerate to transmit based on input resolution/framerate, available bandwidth and whether this source is configured for motion or sharpness. For motion sources, this setting can be used to adjust how rapidly the system should increase the transmitted resolution when increasing bandwidth. For instance, if the source is a camera input is 1920x1080p60, the system will transmit 1920x720p60 at about 2.2Mb/sec and above with this setting set to normal, but will transmit 1920x720p60 at about 1.1Mb/sec and above with this setting set to high. In general you could choose medium or high if you have good quality video inputs and good lighting conditions, while normal would give a good experience in most conditions, including not optimal lighting.

Valuespace: <Normal/Medium/High>

See Table 1: Optimal definition for systems supporting 1080p and Table 2: Optimal definition for systems supporting 720p60.

Example: xConfiguration Video Input Source 1 OptimalDefinition Profile: Normal

<table>
<thead>
<tr>
<th>Table 1: Optimal definition, for systems supporting 1080p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
</tr>
<tr>
<td>Normal</td>
</tr>
<tr>
<td>Medium</td>
</tr>
<tr>
<td>High</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2: Optimal definition, for systems supporting 720p60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
</tr>
<tr>
<td>Normal</td>
</tr>
<tr>
<td>Medium</td>
</tr>
<tr>
<td>High</td>
</tr>
</tbody>
</table>
The Video settings, continued...

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

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

Valuespace: `<512_288/768_448/1024_576/1280_720/Never>

- 512_288: Set the threshold to 512x288.
- 768_448: Set the threshold to 768x448.
- 1024_576: Set the threshold to 1024x576.
- 1280_720: Set the threshold to 1280x720.
- Never: Do not set a threshold for transmitting 60fps.

Example: `xConfiguration Video Input Source 1 OptimalDefinition Threshold60fps: 1280_720`

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

When encoding and transmitting video there will be a 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.

Valuespace: `<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 ScaleToFit**

Defines 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).

Valuespace: `<Manual/MaintainAspectRatio/StretchToFit>

- Manual: If the difference in aspect ratio between the video input source and the target image frame is less than the ScaleToFit configuration (in percent), the image is stretched to fit. Unless 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.

Example: `xConfiguration Video Layout ScaleToFit: MaintainAspectRatio`

**xConfiguration Video Layout ScaleToFitThreshold**

Only applicable if the ScaleToFit 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 ScaleToFitThreshold configuration (in percent), the image is stretched to fit. Unless the system will maintain the original aspect ratio.

Valuespace: `<0..100>

Example: `xConfiguration Video Layout ScaleToFitThreshold: 5`

**xConfiguration Video Layout Scaling**

Defines whether the system should automatically adjust aspect ratio for images or frames when it differs between the image and the frame it is to be placed in.

Valuespace: `<On/Off>

Example: `xConfiguration Video Layout Scaling: On`

**xConfiguration Video MainVideoSource**

Define which video input source shall be used as the main video source. The input source is configured to a video input connector.

Valuespace: `<1..3>

Example: `xConfiguration Video MainVideoSource: 1`

**xConfiguration Video Monitors**

The codec can be used with more than one monitor and this setting lets you set the codec’s monitor layout mode to single, dual, or presentation only on the dual monitor.

Valuespace: `<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`
The Video settings, continued...

**xConfiguration Video OSD InputMethod InputLanguage**
The codec can be enabled for Cyrillic input characters in the GUI (Graphical User Interface). NOTE: Requires that xConfiguration Video OSD inputMethod Cyrillic is set to On.

Valuespace: `<Latin/Cyrillic>`
- **Latin**: Latin characters can be entered using the remote control (default setting).
- **Cyrillic**: Cyrillic characters can be entered using the remote control. NOTE: Requires a TANDBERG Remote Control TRC5 with Cyrillic fonts.

Example: `xConfiguration Video OSD InputMethod InputLanguage: Latin`

**xConfiguration Video OSD InputMethod Cyrillic**
This configuration is used to hide or show the Cyrillic mode as menu input language in the GUI (Graphical User Interface).

Valuespace: `<On/Off>`
- **On**: Cyrillic mode is available as a menu input language in the GUI. This will enable the setting `xConfiguration Video OSD InputMethod InputLanguage`.
- **Off**: Cyrillic mode is NOT available as a menu input language in the GUI.

Example: `xConfiguration Video OSD InputMethod Cyrillic: Off`

**xConfiguration Video OSD Mode**
The Video OSD (On Screen Display) Mode lets you define whether or not information and icons on screen should be displayed.

Valuespace: `<On/Off>`
- **On**: Set to On to display the on screen menus, icons and indicators.
- **Off**: Set to Off to hide the on screen menus, icons and indicators.

Example: `xConfiguration Video OSD Mode: On`

**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 output is displayed on the first monitor. If you have a single monitor and you cannot see any OSD on the connected monitor, if the OSD output has been set to the second monitor and the second monitor is not connected, then you cannot see any menus, information or icons on the screen. To move the OSD output to the first monitor press the following shortcut sequence on the remote control: Press the Disconnect key followed by: * # * # 0 x # (where x is output 1 or 2).

Valuespace: `<1..2>`
- **Range**: Select 1 for HDMI 1 output, or select 2 for DVI-I 2 output.

Example: `xConfiguration Video OSD Output: 1`

The Video settings, continued...

**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 TMS (TANDBERG Management Suite).

Valuespace: `<On/Off>`
- **On**: Displays information about this systems bookings on screen.
- **Off**: Do not display todays bookings.

Example: `xConfiguration Video OSD TodaysBookings: Off`

**xConfiguration Video Output HDMI [1] MonitorRole**
The monitor role describes what video stream will be shown on the monitor connected to the output connector. Applicable only if the “Video > Monitors” configuration is set to dual.

Valuespace: `<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 TV’s 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.

Valuespace: `<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 HDMI video output. This will force the selected resolution on the monitor.

Valuespace: `<Auto/640_480_60/800_600_60/1024_768_60/1280_1024_60/1280_720_60/1920_1080_60/1280_768_60/1360_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, 1280x720@60p, 1920x1080@60p, 1360x768@60p, 1366x768@60p, 1600x1200@60p, 1920x1200@60p

Example: `xConfiguration Video Output HDMI 1 Resolution: 1920_1080_60`
The Video settings, continued...

**xConfiguration Video Output DVI [2] MonitorRole**

The monitor role describes what video stream will be shown on the monitor connected to the output connector. Applicable only if the “Video > Monitors” configuration is set to dual.

Valuespace: <First/Second/PresentationOnly/Third/Fourth>

- **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 TV’s 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.

Valuespace: <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 DVI-I video output. This will force the selected resolution on the monitor.

Valuespace: <Auto/640_480_60/800_600_60/1024_768_60/1280_1024_60/1280_720_60/1920_1080_60/1280_768_60/1360_768_60/1366_768_60/1600_1200_60/1920_1200_60/1280_768_60/1360_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, 1366x768@60p, 1600x1200@60p, 1920x1200@60p

**Example:** `xConfiguration Video Output DVI 2 Resolution: 1024_768_60`

The Video settings, continued...

**xConfiguration Video Output Composite [3] MonitorRole**

NOTE: This command is not supported on Codec C40.

The monitor role describes what video stream will be shown on the monitor connected to the output connector. Applicable only if the monitor configuration is set to dual or quadruple.

Valuespace: <First/Second/PresentationOnly/Third/Fourth>

- **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.
- **Third:** Use for remote participants (only for quadruple monitor setup).
- **Fourth:** Use for remote participants (only for quadruple monitor setup).

**Example:** `xConfiguration Video Output Composite 3 MonitorRole: First`

**xConfiguration Video Output Composite [3] OverscanLevel**

NOTE: This command is not supported on Codec C40.

Some TV’s 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.

Valuespace: <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 video output composite 3. This will force the selected resolution on the monitor.

Valuespace: <PAL/NTSC>

- **Range:** Select PAL or NTSC resolution.

**Example:** `xConfiguration Video Output Composite 3 Resolution: NTSC`
The Video settings, continued...

xConfiguration Video Selfview
The Video Selfview setting determines whether or not the main video source (selfview) should be displayed on screen.

Valuespace: <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
The Video Wallpaper setting determines whether or not a background picture should be displayed on screen when idle.

Valuespace: <None/Growing/Summersky/Custom>
- None: Set to None if you do not want a wallpaper to be displayed on screen.
- Summersky, Growing: Select the wallpaper to be displayed on screen.
- Custom: The custom wallpaper is uploaded from the web interface. Open a web browser and enter the IP address of the codec. Select “Wallpaper” from the menu on top of the screen. Browse for the file and press the “Upload” button. Toggle once between “None” and “Custom” wallpaper to make the change take effect. See the Appendices section for further details about the web interface.

Example: xConfiguration Video Wallpaper: Summersky
The Experimental menu

The Advanced configurations menu has an option called Experimental. The settings within this menu can be used ‘as is’. Most of the settings within the Experimental menu will not be documented.

NOTE! The Experimental menu WILL change.

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

**EchoControl**

- HighPassFilter

The Experimental settings can be used ‘as is’ and will not be further documented. NOTE: The Experimental settings WILL change.

Valuespace: `<On/Off>`

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

**xConfiguration Experimental CapsetFilter**

The Experimental settings can be used ‘as is’ and will not be further documented. NOTE: The Experimental settings WILL change.

Valuespace: `<S: 0, 32>`

Example: `xConfiguration Experimental CapsetFilter: ""`

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

**PacketLossResilience**

The Experimental settings can be used ‘as is’ and will not be further documented. NOTE: The Experimental settings WILL change.

Valuespace: `<On/Off>`

Example: `xConfiguration Experimental Conference 1 PacketLossResilience: Off`

**xConfiguration Experimental CustomSoftbuttons State [1..2]**

**Softbutton [1..5]**

**Type**

The Experimental settings can be used ‘as is’ and will not be further documented. NOTE: The Experimental settings WILL change.

Valuespace: `<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**

The Experimental settings can be used ‘as is’ and will not be further documented. NOTE: The Experimental settings WILL change.

Valuespace: `<S: 0, 255>`

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

**xConfiguration Experimental SoftwareUpgrade Mode**

The Experimental settings can be used ‘as is’ and will not be further documented. NOTE: The Experimental settings WILL change.

Valuespace: `<Auto/Manual>`

Example: `xConfiguration Experimental SoftwareUpgrade Mode: Auto`

**xConfiguration Experimental SoftwareUpgrade ServerAddress**

The Experimental settings can be used ‘as is’ and will not be further documented. NOTE: The Experimental settings WILL change.

Valuespace: `<S: 0, 255>`

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

**xConfiguration Experimental UserInterfaces OSD PhonebookMerged**

The Experimental settings can be used ‘as is’ and will not be further documented. NOTE: The Experimental settings WILL change.

Valuespace: `<On/Off>`

Example: `xConfiguration Experimental UserInterfaces OSD PhonebookMerged: Off`
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.

The Audio commands

---

**xCommand Audio Equalizer Update**

Configure the equalizer parameters by specifying filter type, frequency, gain and Q-value for the selected equalizer (1 .. 8) and the selected section (1 ... 6).

**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:**

```
xCMD xCommand Audio Equalizer Update
  EqualizerId: 1
  Section: 1
  FilterType: "Peaking"
  Frequency: "100"
  Q: "4"
  Gain: "0"
```

---

**xCommand Audio Equalizer List**

Shows the current equalizer settings for the codec.

**Parameters:**

- **EqualizerId:** <1..8>

**Example:**

```
xCMD xCommand Audio Equalizer List
  EqualizerId: 1
```

---

```none
*r AudioEqualizerListResult Equalizer 1 Section 1 FilterType: "Peaking"
*r AudioEqualizerListResult Equalizer 1 Section 1 Frequency: "100"
*r AudioEqualizerListResult Equalizer 1 Section 1 Q: "4"
*r AudioEqualizerListResult Equalizer 1 Section 1 Gain: "0"
*r AudioEqualizerListResult Equalizer 1 Section 2 FilterType: "Peaking"
*r AudioEqualizerListResult Equalizer 1 Section 2 Frequency: "200"
*r AudioEqualizerListResult Equalizer 1 Section 2 Q: "4"
*r AudioEqualizerListResult Equalizer 1 Section 2 Gain: "0"
*r AudioEqualizerListResult Equalizer 1 Section 3 FilterType: "Peaking"
*r AudioEqualizerListResult Equalizer 1 Section 3 Frequency: "500"
*r AudioEqualizerListResult Equalizer 1 Section 3 Q: "4"
*r AudioEqualizerListResult Equalizer 1 Section 3 Gain: "0"
*r AudioEqualizerListResult Equalizer 1 Section 4 FilterType: "Peaking"
*r AudioEqualizerListResult Equalizer 1 Section 4 Frequency: "1000"
*r AudioEqualizerListResult Equalizer 1 Section 4 Q: "4"
*r AudioEqualizerListResult Equalizer 1 Section 4 Gain: "0"
*r AudioEqualizerListResult Equalizer 1 Section 5 FilterType: "Peaking"
*r AudioEqualizerListResult Equalizer 1 Section 5 Frequency: "2000"
*r AudioEqualizerListResult Equalizer 1 Section 5 Q: "4"
*r AudioEqualizerListResult Equalizer 1 Section 5 Gain: "0"
*r AudioEqualizerListResult Equalizer 1 Section 6 FilterType: "Peaking"
*r AudioEqualizerListResult Equalizer 1 Section 6 Frequency: "5000"
*r AudioEqualizerListResult Equalizer 1 Section 6 Q: "0"
*r AudioEqualizerListResult Equalizer 1 Section 6 Gain: "0"
```
** end
The Audio commands, cont...

**xCommand Audio Microphones Mute**
Mute all microphones.

**Parameters:**
No parameters.

**Example:**
```
xCommand Audio Microphones Mute
*r AudioMicrophonesMuteResult (status=OK):
** end
```

**xCommand Audio Microphones Unmute**
Unmute microphones.

**Parameters:**
No parameters.

**Example:**
```
xCommand Audio Microphones Unmute
*r AudioMicrophonesUnmuteResult (status=OK):
** end
```

**xCommand Audio Sound Play**
Play the specified audio sound.

**Parameters:**

- **Sound(r):** <Busy/CallWaiting/Dial/KeyTone/Ringing/SpecialInfo/TelephoneCall/VideoCall>
- **Loop:** <On/Off>

**Example:**
```
xCommand Audio Sound Play Sound: Ringing
*r AudioSoundPlayResult (status=OK):
** end
```

**xCommand Audio Sound Stop**
Stop playing audio sound.

**Parameters:**
No parameters.

**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.

**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.

**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 Audio commands, cont...

**xCommand Audio Setup Clear**
Removes all local inputs and local outputs.

**Parameters:**
No parameters.

**Example:**
```
xCommand Audio Setup Clear
`r AudioSetupClearResult (status=OK):
** end
```

**xCommand Audio LocalInput Add**
Creates a local input and generates the local input id. A local input is a mix of input connectors with the following settings: Name, MixerMode, AGC, Mute and Channels.

- **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.
  - 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.
- **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.

**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 Remove**
Removes the local input given by the input ID.

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

**Example:**
```
xCommand Audio LocalInput Remove InputID: 2
OK
`r AudioInputGroupRemoveResult (status=OK):
** end
```

**xCommand Audio LocalInput Update**
Updates the settings of the local input given by the input ID.

- **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.
  - 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.
- **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.

**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):
InputId: 2
** end
```
The Audio commands, cont...

**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.

### 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.

### 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
```

The Audio commands, cont...

**xCommand Audio LocalOutput Add**
Creates a local output and generates 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.

A local output has the following settings:
- **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.

### 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
```
The Audio commands, cont...

**xCommand Audio LocalOutput Update**
Updates the settings of the local output given by the output ID.

*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.

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

**Example:**
```
xCommand Audio LocalOutput Update OutputId: 5 Name: "Loudspeaker"  
Loudspeaker: On Channels: 2  
OK  
*r AudioOutputGroupUpdateResult (status=OK):  
  ** end
```

**xCommand Audio LocalOutput AddConnector**
Attach an output connector to the local output given by the output ID. A connector is defined by its type and ID.

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

**Example:**
```
xCommand Audio LocalOutput AddConnector OutputId:5 ConnectorType: Line  
ConnectorId:1  
OK  
*r AudioOutputGroupAddConnectorResult (status=OK):  
  ** end
```

**xCommand Audio LocalOutput RemoveConnector**
Detach an output connector from the local output given by the output ID. A connector is defined by its type and ID.

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

**Example:**
```
xCommand Audio LocalOutput RemoveConnector OutputId:5 ConnectorType: Line  
ConnectorId:1  
OK  
*r AudioOutputGroupRemoveConnectorResult (status=OK):  
  ** end
```

*xCommand Audio LocalOutput Remove*  
Removes the local output given by the output ID.

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

**Example:**
```
xCommand Audio LocalOutput Remove OutputId: 6  
OK  
*r AudioOutputGroupRemoveResult (status=OK):  
  ** end
```
The Audio commands, cont...

**xCommand Audio LocalOutput ConnectInput**
Connect a local input or remote input to a local output by giving their IDs as parameters. If desired, set a gain on the input signal in the range from -53 dB to 15 dB. -54 dB equals OFF.

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

**Example:**
```
xCommand Audio LocalOutput ConnectInput OutputId:6 InputId:3
OK
*  AudioOutputGroupConnectInputResult (status=OK):
** end
```

**xCommand Audio LocalOutput UpdateInputGain**
Update the gain of a local input or remote input connected to a local output. The gain on the input signal is in the range from -53dB to 15dB. The value -54dB equals Off. See the Appendix > Dynamic audio API section.

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

**Example:**
```
xCommand Audio LocalOutput UpdateInputGain OutputId:6 InputId:3
OK
*  AudioOutputGroupUpdateInputGainResult (status=OK):
** end
```

**xCommand Audio LocalOutput DisconnectInput**
Disconnect a local input or remote input from a local output by giving their IDs as parameters.

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

**Example:**
```
xCommand Audio LocalOutput DisconnectInput OutputId:6 InputId:3
OK
*  AudioOutputGroupDisconnectInputResult (status=OK):
** end
```

**xCommand Audio RemoteInput Update**
When a call is made a remote input and remote output pair is created. This command updates the settings of the remote input given by the input ID. The remote input has one setting, the AGC (Automatic Gain Control).

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

**Example:**
```
xCommand Audio RemoteInput Update InputId:9 AGC: Off
OK
*  AudioRemoteInputGroupUpdateResult (status=OK):
** end
```
The Audio commands, cont...

**xCommand Audio RemoteOutput ConnectInput**
When a call is made a remote input and remote output pair is created. This command connects a local input or remote input to a remote output with their IDs as parameters. If desired, set a gain on the input signal in the range from -53 dB to 15 dB. -54 dB equals OFF.

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 input or remote input connected to a remote output. The gain on the input signal is in the range from -53dB to 15dB. The value -54dB equals Off. See the Appendix > Dynamic audio API section.

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
```

The Boot command

**xCommand Boot**
Reboot system.

Parameters:
- No parameters.

Example:
```
xCommand Boot
*r BootResult (status=OK):
** endOK
CUIL reboot request, restarting
Connection closed by foreign host.
```
The Call commands

**xCommand Call Disconnect**  
Disconnect specified call. (Tip: xstats Call will give you a list of all active calls)

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

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

**xCommand Call DisconnectAll**  
Disconnect all active calls.

**Parameters:**
- No parameters.

**Example:**
```
xCommand Call DisconnectAll
OK
*r DisconnectAllResult (status=OK):
** end
```

**xCommand Call Accept**  
Accept incoming call. If no call id is specified, all incoming calls will be accepted.

**Parameters:**
- CallId: <0..65534>

**Example:**
```
xCommand Call Accept CallId:19
OK
*r CallAcceptResult (status=OK):
** end
```

**xCommand Call Reject**  
Reject incoming call. If no call id is specified, all incoming calls will be rejected.

**Parameters:**
- CallId: <0..65534>

**Example:**
```
xCommand Call Reject CallId:20
OK
*r CallRejectResult (status=OK):
** end
```
The Call Log 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, all call logs will be deleted.

**Parameters:**
LogTag: <0..2147483647>

**Example:**

```
xCommand CallLog Clear
*r ClearResult (status=OK):
** end
```

**xCommand CallLog Recent Delete**
Delete 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 list will be deleted.

**Parameters:**
LogTag: <0..2147483647>

**Example:**

```
xCommand CallLog Recent Delete
*r DeleteResult (status=OK):
** end
```

**xCommand CallLog Outgoing Delete**
Delete log of outgoing calls made from the system. If a logtag is given as argument, that specific call is deleted from the log. If no logtag is given, the complete outgoing calls list will be deleted.

**Parameters:**
LogTag: <0..2147483647>

**Example:**

```
xCommand CallLog Outgoing Delete LogTag:202
*r DeleteResult (status=OK):
** end
```

The Call Log commands, cont...

**xCommand CallLog Received Delete**
Delete log of received calls. If a logtag is given as argument, that specific call is deleted from the log. If not logtag is given, the complete received calls list will be deleted.

**Parameters:**
LogTag: <0..2147483647>

**Example:**

```
xCommand CallLog Received Delete LogTag:126
*r DeleteResult (status=OK):
** end
```

**xCommand CallLog Missed Delete**
Delete log of missed calls. If a logtag is given as argument, that specific call is deleted from the log. If not logtag is given, the complete missed calls list will be deleted.

**Parameters:**
LogTag: <0..2147483647>

**Example:**

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

xCommand CamCtrlPip
Can be used to show or hide a small window with the camera selfview, while adjusting the camera position for instance. (If selfview isn't already shown)

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

Example:
  xCommand CamCtrlPip Mode: On
  *r CamCtrlPipResult (status=OK):
  ** end

The Camera commands

xCommand Camera PanTiltReset
Camera pan/tilt values is positioned to its default predefined value.

Parameters:
CameraId(r): <1..7>

Example:
  xCommand Camera PanTiltReset CameraId:1
  OK
  *r PanTiltResetResult (status=OK):
  ** end

xCommand Camera PositionSet
Set camera position to specified values.

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
The Camera commands, cont...

**xCommand Camera Ramp**
Move camera in specified direction. Camera will move at specified speed until stop command is issued. Run a stop command to stop the camera:

**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 ReconfigureCameraChain**
Reinitialize camera chain. Get info about what camera is sitting in what position in the camera chain.

**Parameters:**
- No parameters.

**Example:**
```
xCommand Camera ReconfigureCameraChain
*r ReconfigureCameraChainResult (status=OK):
** end
```

The Camera commands, cont...

**xCommand Camera TriggerAutofocus**
Trigger the automatic focus functionality in the selected camera.

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

**Example:**
```
xCommand Camera TriggerAutofocus CameraId:1
OK
*r TriggerAutofocusResult (status=OK):
** end
```

The Dial command

**xCommand Dial**
Dial out from the system.

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

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

The DTMF command

**xCommand DTMFSend**
Send DTMF tones to the far end.

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

**Example:**
```plaintext
xCommand DTMFSend CallId:2 DTMFString:1234
*r DTMFSendResult (status=OK):
** end
```

The Far End Control commands

**xCommand FarEndControl Camera Move**
Move far end camera. The camera will move in the specified direction until the stop command is issued.

**Parameters:**
- CallId(r): <0..65534>
- Value(r): <Left/Right/Up/Down/ZoomIn/ZoomOut>

**Example:**
```plaintext
xCommand FarEndControl Camera Move CallId:3 Value:left
*r FECCMoveResult (status=OK):
** end
```

**xCommand FarEndControl Camera Stop**
Stop far end camera control move.

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

**Example:**
```plaintext
xCommand FarEndControl Camera Stop CallId:3
*r FECCMoveResult (status=OK):
** end
```

**xCommand FarEndControl Preset Activate**
Activate move of far end camera to preset position.

**Parameters:**
- CallId(r): <0..65534>
- PresetId(r): <1..15>

**Example:**
```plaintext
xCommand FarEndControl Preset Activate CallId:3 PresetId:1
*r FECCPresetActivateResult (status=OK):
** end
```
The Far End Control commands, cont...

xCommand FarEndControl Preset Store
Store position of far end camera to a preset.

Parameters:
- CallId(r): <0..65534>
- PresetId(r): <0..15>

Example:
```
xCommand FarEndControl Preset Store CallId:3 PresetId:1
*r FECCPresetStoreResult (status=OK):
** end
```

xCommand FarEndControl Source Select
Select which video source to use as the main source on the far end system.

Parameters:
- CallId(r): <0..65534>
- SourceId(r): <0..15>

Example:
```
xCommand FarEndControl Source Select CallId:3 SourceId:1
*r FECCSelectSourceResult (status=OK):
** end
```

The GPIO commands

xCommand GPIO ManualState Set
NOTE: This command is not supported on Codec C40.
Command used to set a GPIO pin that is configured to OutputManualState mode.

Parameters:
- Pin1: <High/Low>
- Pin2: <High/Low>
- Pin3: <High/Low>
- Pin4: <High/Low>

Example:
```
xCommand GPIO ManualState Set Pin1: High
*r GpioStateSetResult (status=OK):
** end
```
The HTTP Feedback commands

**xCommand HttpFeedback Register**
Command used to instruct the system to return XML feedback over HTTP(S) to specific URLs. What parts of the Status and Configuration XML documents to monitor are specified by XPath expressions.

**Parameters:**
- FeedbackSlot: <1..3>
- ServerUrl: <S: 1, 256>
- Expression[1..15]: <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
```

**xCommand HttpFeedback Deregister**
Command used to deregister XML feedback over HTTP(S).

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

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

The Key commands

**xCommand Key Click**
Command used to emulate pressing a key on the remote control for a short while. This command needs no release command.

**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
- Duration: <0..65534>

**Example:**
```
xCommand Key Click Key:Down Duration:1
```

**xCommand Key Press**
Command used to emulate pressing a key on the remote control without releasing it. The Key Press command should be followed by a Key Release command to emulate releasing the key.

**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:Home
```

**xCommand Key Release**
Command used to emulate releasing an already pressed key on the remote control. The Key Release command should be preceded by a Key Press command to emulate pressing the key.

**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
```
The Message command

**xCommand Message Alert Display**

This command will make the system display a window on the graphical user interface with the provided text.

Parameters:
- Text(r): <S: 0, 255>
- Duration: <0..3600>

Example:

```
xCommand Message Alert Display
Text: "The meeting will end in 5 minutes."
Duration: 20
OK
*r MessageAlertDisplayResult (status=OK):
** end
```

**xCommand Message Alert Clear**

This command will remove the window displayed using the xCommand Message Alert Display command.

Parameters:
- No parameters.

Example:

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

**xCommand Message Prompt Display**

This command will make the system display a window on the graphical user interface. The window will contain of a text together and possible responses from the user.

Parameters:
- Title: <S: 0, 255>
- Text(r): <S: 0, 255>
- FeedbackId: <S: 0, 255>
- Option.1: <S: 0, 255>
- Option.2: <S: 0, 255>
- Option.3: <S: 0, 255>
- Option.4: <S: 0, 255>
- Option.5: <S: 0, 255>

Example:

```
xCommand Message Prompt Display
Title: "Meeting extension"
Text: "The meeting is about to end. Do you want to extend the meeting?"
Option.1: "No"
Option.2: "Yes, extend with 5 minutes"
Option.3: "Yes, extend with 10 minutes"
OK
*r MessagePromptDisplayResult (status=OK):
*r/end
```

**xCommand Message Prompt Clear**

This command will remove the window displayed using the xCommand Message Alert Display command.

Parameters:
- FeedbackId: <S: 0, 255>

Example:

```
xCommand Message Prompt Clear
OK
*r MessagePromptClearResult (status=OK):
*r/end
```
The Message commands, cont...

**xCommand Message Prompt Response**
This command can be used as a response to the command xCommand Message Prompt Display. The optionId corresponds to the optionIds given as possible responses in the Display command.

**Parameters:**
- FeedbackId: <S: 0, 255>
- OptionId(r): <0..5>

**Example:**
```
xCommand Message Prompt Response OptionId: 1
OK
*r MessagePromptResponseResult (status=OK):
*r/end
```

The Phonebook commands

**xCommand Phonebook Contact Add**
Add new contact to local phonebook. Stored internally in system.

**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>

**Example:**
```
xCommand Phonebook Contact Add Name:"John Doe" Number:12345
OK
*r PhonebookContactAddResult (status=OK):
Name: localContactId-1
** end
```

**xCommand Phonebook Contact Modify**
Modify existing contact in local phonebook.

**Parameters:**
- ContactId(r): <S: 0, 255>
- Name: <S: 0, 255>
- FolderId: <S: 0, 255>
- ImageURL: <S: 0, 255>
- Title: <S: 0, 255>

**Example:**
```
xCommand Phonebook Contact Modify ContactId:localContactId-1 Name:"John Doe - office"
OK
*r PhonebookContactModifyResult (status=OK):
** end
```
The Phonebook commands, cont...

**xCommand Phonebook Contact Delete**
Delete contact from local phonebook.

**Parameters:**
- ContactId(r): <S: 0, 255>

**Example:**
```
xCommand Phonebook Contact Delete ContactId:localContactId-1
OK
```

**r PhonebookContactDeleteResult (status=OK):**
** end

**xCommand Phonebook ContactMethod Add**
Specify details about how to set up call to local phonebook contact.

**Parameters:**
- ContactId(r): <S: 0, 255>
- Device: <Mobile/Other/Telephone/Video>
- Number(r): <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 ContactMethod Delete**
Delete call setup details for local phonebook contact.

**Parameters:**
- ContactId(r): <S: 0, 255>
- ContactMethodId(r): <S: 0, 255>

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

**r PhonebookContactMethodDeleteResult (status=OK):**
** end

**xCommand Phonebook Folder Add**
Add a folder to the local phonebook, where phonebook entries can be stored.

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

**Example:**
```
xCommand Phonebook Folder Add Name: locationA
OK
```

**r PhonebookFolderAddResult (status=OK):**
```
Name: localGroupId-3
** end
```

**xCommand Phonebook Folder Modify**
Modify an existing phonebook folder.

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

**Example:**
```
xCommand Phonebook Folder Modify FolderId:localGroupId-3 Name:locationB
OK
```

**r PhonebookFolderModifyResult (status=OK):**
** end
The Phonebook commands, cont...

**xCommand Phonebook Folder Delete**
Delete an existing folder from the local phonebook.

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

**Example:**
```
xCommand Phonebook Folder Delete FolderId:localGroupId-3
OK
*r PhonebookFolderDeleteResult (status=OK):
** end
```

**xCommand Phonebook Search**
Search for specific entries in the phonebook.

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

**Example:**
```
xCommand Phonebook Search PhonebookId: departmentA PhonebookType: Corporate
SearchString: John SearchField: Name
*r PhonebookSearchResult (status=OK):
  Name: 6
** end
```

The Presentation commands

**xCommand Presentation Start**
Will open dual presentation stream using selected presentation source when in call (if presenter option is installed). When not in call, the selected presentation source will be displayed.

**Parameters:**
- PresentationSource: <1..5>

**Example:**
```
xCommand Presentation Start PresentationSource:2
OK
*r PresentationStartResult (status=OK):
** end
```

**xCommand Presentation Stop**
Will stop dual presentation stream when in call. Will stop displaying presentation source locally when not in call.

**Parameters:**
- No parameters.

**Example:**
```
xCommand Presentation Stop
OK
*r PresentationStopResult (status=OK):
** end
```
The Preset commands

**xCommand Preset Activate**
Activate local preset. Will move camera to predefined position.

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

**Example:**
```
xCommand Preset Activate PresetId:3
OK
```
```
*r PresetActivateResult (status=OK):
** end
```

**xCommand Preset Clear**
delete a preset storing a camera position.

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

**Example:**
```
xCommand Preset Clear PresetId:3
OK
```
```
*r PresetClearResult (status=OK):
** end
```

**xCommand Preset Store**
Store current camera position. System may hold 15 predefined camera positions (presets).

**Parameters:**
- PresetId(r): <1..15>
- Type(r): <All/Camera>
  - Description: <S: 0, 255>

**Example:**
```
xCommand Preset Store PresetId:3 Type:Camera Description:"Left view"
OK
```
```
*r PresetStoreResult (status=OK):
** end
```

The SString Send command

**xCommand SStringSend**
Used to send data to far end, e.g. for control systems. Uses the H.224 data channel (UDP).

**Parameters:**
- Message(r): <S: 1, 256>
- CallId: <0..65534>

**Example:**
```
xCommand SStringSend CallId:4 Message:"This is a test"
```
```
*r SStringSendResult (status=OK):
** end
```
The Standby command

**xCommand Standby Activate**
Set system in standby mode, which will turn off the video outputs and put the camera to sleep.

- **Parameters:** No parameters.
- **Example:**
  ```
  xCommand Standby Activate
  *r ActivateResult (status=OK):
  ** end
  ```

**xCommand Standby Deactivate**
Bring system out of standby mode.

- **Parameters:** No parameters.
- **Example:**
  ```
  xCommand Standby Deactivate
  *r DeactivateResult (status=OK):
  ** end
  ```

**xCommand Standby ResetTimer**
System will enter standby mode after time specified by Delay parameter.

- **Parameters:** Delay: <1..480>
- **Example:**
  ```
  xCommand Standby ResetTimer Delay:10
  *r ResetResult (status=OK):
  ** end
  ```

The SystemUnit command

**xCommand SystemUnit OptionKey Add**
Add option key to add extra system capabilities, e.g. Multisite.

- **Parameters:**
  - Key(r): <S: 16, 24>
- **Example:**
  ```
  xCommand SystemUnit OptionKey Add Key:******************
  *r OptionKeyResult (status=OK):
  ** end
  ```

**xCommand SystemUnit ReleaseKey Add**
Add software release key. Used to enable new software (applicable for main sw releases).

- **Parameters:**
  - Key(r): <S: 16, 24>
- **Example:**
  ```
  xCommand SystemUnit ReleaseKey Add Key:******************
  *r ReleaseKeyResult (status=OK):
  ** end
  ```

**xCommand SystemUnit AdminPassword Set**
Set administrator password to access the codec.

- **Parameters:**
  - Password(r): <S: 0, 255>
- **Example:**
  ```
  xCommand SystemUnit AdminPassword Set Password:***********
  *r AdminPasswordSetResult (status=OK):
  ** end
  ```
**xCommand SystemUnit MenuPassword Set**
Set password needed to access the Advanced menu settings.

**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.

**Parameters:**
- Password(r): <S: 0, 255>

**Example:**
```
xCommand SystemUnit MenuPassword Validate Password:**********
*r: MenuPasswordValidateResult (status=OK):
  ** end
```

**xCommand SystemUnit DateTime Set**
Configure the date and time into the system, if not available from NTP.

**Parameters:**
- Year: <2008..2037>
- Month: <1..12>
- Day: <1..31>
- Hour: <0..23>
- Minute: <0..59>
- Second: <0..59>

**Example:**
```
xCommand SystemUnit DateTime Set Year:2009 Month:7 Day:3 Hour:12 Minute:0 Second:0
*r: DateTimeSetResult (status=OK):
  ** end
```

**xCommand SystemUnit DateTime Get**
Read out the time end date from the system

**Parameters:**
- No parameters.

**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 all the layout compositions to factory default settings.

**Parameters:**
- Confirm(r): <Yes>

**Example:**
```
xCommand SystemUnit FactoryReset Confirm: Yes
*r: FactoryResetConfirmResult (status=OK):
  ** end
```

**xCommand SystemUnit ConfigurationProfile SaveCurrentConfigurationAs**
Save a copy of the current configuration profile and assign a name to it.

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

**Example:**
```
xCommand SystemUnit ConfigurationProfile SaveCurrentConfigurationAs Name: "My _ ConfigurationProfile _1"
*r: ConfigurationProfileSaveCurrentConfigurationResult (status=OK):
  ** end
```
The System Unit commands, cont...

**xCommand SystemUnit ConfigurationProfile Change**
Select a previously saved configuration profile. Will be active after next system boot.

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

**Example:**
```plaintext
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 CancelChange**
Cancel the "ConfigurationProfile Change" command, that would otherwise take effect after next system boot.

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

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

**xCommand SystemUnit ConfigurationProfile List**
List configuration profiles that has been stored in the system.

**Parameters:**
- No parameters.

**Example:**
```plaintext
xCommand SystemUnit ConfigurationProfile List
```
```
*r ConfigurationProfileListResult (status=OK):
    Profile: My_ConfigurationProfile_1
    Profile: My_ConfigurationProfile_2
** end
```

**xCommand SystemUnit ConfigurationProfile Remove**
Delete a configuration profile that has been stored in the system.

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

**Example:**
```plaintext
xCommand SystemUnit ConfigurationProfile Remove Name: "My_ConfigurationProfile_1"
```
```
*r ConfigurationProfileRemoveResult (status=OK):
** end
```
The TStringSend commands

**xCommand TStringSend**
Used to send data to far end, e.g. for Telepresence control systems. Uses the H.245 control channel (TCP). Only works with H.323 calls.

**Parameters:**
- CallId(r): <0..65534>
- Message(r): <S: 1, 1450>

**Example:**
```
xCommand TStringSend CallId:1 Message:"This is an example"
*r TStringSendResult (status=OK):
  ** end
```

The Video commands

**xCommand Video PictureLayoutSet**
Select screen layout mode.

**Parameters:**
- LayoutFamily(r): <Auto/Equal/Fullscreen/PresentationLargeSpeaker/PresentationSmallSpeaker>

**Example:**
```
xCommand Video PictureLayoutSet LayoutFamily: Equal
*r SetPictureLayoutResult (status=OK):
  ** end
```

**xCommand Video Layout AutoMode SetLayoutFamily**
Set the default picture layout for the specified situation.

**Parameters:**
- Monitors(r): <Dual/Single>
- Selfview(r): <On/Off>
- Presentation(r): <16_9/4_3/Off>
- NumberOfCalls(r): <0..3>
- LayoutFamily(r): <Auto/Equal/Fullscreen/PresentationLargeSpeaker/PresentationSmallSpeaker>

**Example:**
```
xCommand Video Layout AutoMode SetLayoutFamily Monitors: Single Selfview: On Presentation: Off NumberOfCalls: 0 LayoutFamily: Equal
*r VideoLayoutAutoModeSetLayoutFamilyResult (status=OK):
  ** end
```

**xCommand Video Layout AutoMode Reset**
Reset all default picture layouts to the system defaults.

**Parameters:**
- No parameters.

**Example:**
```
xCommand Video Layout AutoMode Reset
*r VideoLayoutAutoModeResetResult (status=OK):
  ** end
```
The Video commands, cont...

**xCommand Video Layout AutoMode List**
Shows the current default picture layouts.

**Parameters:**
No parameters.

**Example:**
xCommand Video Layout AutoMode List
WARN: VideoLayoutAutoModeListResult LayoutAutoMode 1 SingleMonitor 1
SelfviewOff 1 PresentationOff 1 NumberOfCalls 0 LayoutFamily: “fullscreen”
WARN: VideoLayoutAutoModeListResult LayoutAutoMode 1 SingleMonitor 1
SelfviewOff 1 PresentationOff 1 NumberOfCalls 1 LayoutFamily: “fullscreen”
// Comment: Continues with a list of the automode layouts. //
** end

**xCommand Video Layout ListLayoutFamily**
Displays a list of all the defined layout families.

**Parameters:**
LayoutFamilyId: <0..65534>>
DescriptorId: <0..65534>>

**Example:**
xCommand Video Layout ListLayoutFamily
//Comment: Displays the complete list of layout families. //
** end

**xCommand Video Layout ListLayoutGraphic**
Displays a list of all the defined layout graphics.

**Parameters:**
LayoutGraphicId: <0..65534>>
FrameId: <0..65534>>

**Example:**
xCommand Video Layout ListLayoutGraphic
//Comment: Displays the complete list of layout graphics. //
** end

**xCommand Video Layout Add**
NOTE: This command is not supported on Codec C40.
This command will add a new, empty video layout composition. The result will be an id of the newly created layout, which you can use to add frames with different video sources to the layout.

**Parameters:**
LayoutId: <1..2147483647>>

**Example:**
xCommand Video Layout Add
OK
*r VideoLayoutAddResult (status=OK): LayoutId: 1
** end

**xCommand Video Layout Remove**
NOTE: This command is not supported on Codec C40.
Removes an existing video layout.

**Parameters:**
LayoutId(r): <1..2147483647>>

**Example:**
xCommand Video Layout Remove LayoutId: 1
*r VideoLayoutRemoveResult (status=OK):
** end

**xCommand Video Layout RemoveAll**
NOTE: This command is not supported on Codec C40.
Removes all the existing video layouts.

**Parameters:**
No parameters.

**Example:**
xCommand Video Layout RemoveAll
*r VideoLayoutRemoveAllResult (status=OK):
** end
The Video commands, cont...

**xCommand Video Layout Reset**
NOTE: This command is not supported on Codec C40.
Resets all the layout compositions to factory default.

**Parameters:**
No parameters.

**Example:**
```
xCommand Video Layout Reset
```

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

**xCommand Video Layout Frame Add**
NOTE: This command is not supported on Codec C40.
Add a Video frame to an existing layout. Select size and position of the frame, and what videosource that should be shown in the frame.

**Parameters:**
- LayoutId(r): <1..2147483647>
- FrameId: <1..65535>
- PositionX(r): <0..10000>
- PositionY(r): <0..10000>
- Width(r): <1..10000>
- Height(r): <1..10000>
- Layer(r): <1..5>
- Border: <On/Off>
- VideoSourceType(r): <graphic/localInput/localMain/localPresentation/mostSpeaking/otherMain/ownMain/presentation/remoteMain/remotePresentation/videoFile>
- VideoSourceId: <0..256>

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

```
r VideoLayoutFrameAddResult (status=OK):
  FrameId: 1
** end
```

**xCommand Video Layout Frame Remove**
NOTE: This command is not supported on Codec C40.
Remove a video frame from an existing layout.

**Parameters:**
- LayoutId(r): <1..2147483647>
- FrameId(r): <1..65535>

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

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

**xCommand Video Layout Frame Update**
NOTE: This command is not supported on Codec C40.
Change the position or content of an existing frame in a layout.

**Parameters:**
- LayoutId(r): <1..2147483647>
- FrameId(r): <1..65535>
- PositionX(r): <0..10000>
- PositionY(r): <0..10000>
- Width(r): <1..10000>
- Height(r): <1..10000>
- Layer(r): <1..5>
- Border: <On/Off>
- VideoSourceType(r): <Graphic/LocalInput/LocalMain/LocalPresentation/MostSpeaking/OtherMain/OwnMain/Presentation/RemoteMain/RemotePresentation/VideoFile>
- VideoSourceId: <0..256>

**Example:**
```
xCommand Video Layout Frame Update LayoutId:1 FrameId:1 PositionX:200 PositionY:200 Width:9600 Height:9600 Layer:1 Border:off VideoSourceType:localInput VideoSourceId:1
OK
```

```
r VideoLayoutFrameUpdateResult (status=OK):
  ** end
```
The Video commands, cont...

**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.

**Parameters:**

CallId: <0..65534>

OutputId: <0..65534>

LayoutId: <1..2147483647>

**Example:**

xCommand Video Layout Assign Callid:1 OutputId:1 LayoutId:2
OK
*r VideoLayoutAssignResult (status=OK):
  ** end

**xCommand Video Layout AssignLocalOutput**

NOTE: This command is not supported on Codec C40.

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

**Parameters:**

OutputId: <0..65534>

LayoutId: <1..2147483647>

**Example:**

xCommand Video Layout AssignLocalOutput OutputId:1 LayoutId:2
OK
*r VideoLayoutAssignLocalOutputResult (status=OK):
  ** end

**xCommand Video Layout AssignCall**

NOTE: This command is not supported on Codec C40.

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

**Parameters:**

CallId: <0..65534>

LayoutId: <1..2147483647>

**Example:**

xCommand Video Layout AssignCall CallId:1 LayoutId:2
OK
*r VideoLayoutAssignCallResult (status=OK):
  ** end

**xCommand Video Layout AssignPresentation**

NOTE: This command is not supported on Codec C40.

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

**Parameters:**

LayoutId: <1..2147483647>

**Example:**

xCommand Video Layout AssignPresentation LayoutId:2
OK
*r VideoLayoutAssignPresentationResult (status=OK):
  ** end
The Video commands, *cont...

**xCommand Video Layout UnAssign**

NOTE: This command is not supported on Codec C40.
Remove the defined video layout, and go back to default.

**Parameters:**

- CallId(r): <0..65534>
- OutputId(r): <0..65534>

**Example:**

```plaintext
*xCommand Video Layout UnAssign CallId: 1 OutputId: 1
* r VideoLayoutUnassignResult {status=OK}:
** end
```

**xCommand Video Layout UnAssignLocalOutput**

NOTE: This command is not supported on Codec C40.
Remove the defined video layout, and go back to default.

**Parameters:**

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

**Example:**

```plaintext
*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.

**Parameters:**

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

**Example:**

```plaintext
*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.

**Parameters:**

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

**Example:**

```plaintext
*xCommand Video Layout UnAssignPresentation LayoutId:2
OK
* r VideoLayoutUnAssignPresentationResult (status=OK):
** end
```
The Experimental commands

The Experimental commands can be used ‘as is’ and will not be further documented. **NOTE!** The Experimental commands WILL change.

**xCommand Experimental Audio LocalOutput Unmute**

The Experimental commands can be used ‘as is’. **NOTE!** The Experimental settings WILL change. Unmute the local audio output.

**Parameters:**

ConferenceId(r): <0..65534>

**Example:**

```
xCommand Experimental Audio LocalOutput Unmute ConferenceID:1
*r AudioLocalOutputUnmuteResult (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.

The Audio status

xStatus Audio

Run this command to see an overview of the audio status. In the example below we have created one local input (ID 46) and one local output (ID 48). The system is in a call (ID 28). There is one remote input (ID 51) and one remote output (ID 52). The same IDs will be used in the audio examples in the following pages when query status information.

Example:

```
xStatus Audio
*s Audio Microphones Mute: On
*s Audio Volume: 70
*s Audio Input LocalInput 46 Name: "Microphone"
*s Audio Input LocalInput 46 MixerMode: "GainShared"
*s Audio Input LocalInput 46 Mute: "Off"
*s Audio Input LocalInput 46 Channels: 1
*s Audio Input LocalInput 46 AGC: "Off"
*s Audio Input LocalInput 46 Connector: "Microphone.1"
*s Audio Input RemoteInput 51 CallId: 28
*s Audio Input RemoteInput 51 AGC: "Off"
*s Audio Output LocalOutput 48 Name: "MyLocalOutput1"
*s Audio Output LocalOutput 48 Loudspeaker: "Off"
*s Audio Output LocalOutput 48 Channels: 1
*s Audio Output LocalOutput 48 Connector: "Line.1"
*s Audio Output LocalOutput 48 Input 51 Gain: 0
*s Audio Output RemoteOutput 52 CallId: 28
*s Audio Output RemoteOutput 52 Input 46 Gain: 0
*s Audio Module 1 Type: Unknown
*s Audio Module 1 SoftwareID: ""
*s Audio Module 1 HardwareID: ""
*s Audio Module 1 Connector: ""
** end
```

xStatus Audio Microphones Mute

States whether all microphones are muted or unmuted.

Valuespace:

<On/Off>

Example:

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

xStatus Audio Volume

States the volume level of the loudspeaker output.

Valuespace:

<0..100dB>

Example:

```
xStatus Audio Volume
*s Audio Volume: 70
** end
```

xStatus Audio Input LocalInput [1..n] Name

States the name of the local input with the given input ID.

Valuespace:

<S: 0..255>

Example:

```
xStatus Audio Input LocalInput 46 Name
*s Audio Input LocalInput 46 Name: "Microphone"
** end
```

xStatus Audio Input LocalInput [1..n] MixerMode

A local input is a group of input connectors mixed together by the stated MixerMode.

Valuespace:

<Auto/Fixed/GainShared>

Example:

```
xStatus Audio Input LocalInput 46 MixerMode
*s Audio Input LocalInput 46 MixerMode: "Auto"
** end
```
The Audio status, cont...

**xStatus Audio Input LocalInput [1..n] Mute**
States whether the local input with the given input ID is muted.

**Valuespace:**
<On/Off>

**Example:**
```
xStatus Audio Input LocalInput 46 Mute
```
```
*s Audio Input LocalInput 46 Mute: "Off"
** end
```

**xStatus Audio Input LocalInput [1..n] Channels**
States whether the local input mixes the input connectors into a mono signal or a stereo signal.

**Valuespace:**
<1..2>

**Example:**
```
xStatus Audio Input LocalInput 46 Channels
```
```
*s Audio Input LocalInput 46 Channels: 1
** end
```

**xStatus Audio Input LocalInput [1..n] AGC**
States whether Automatic Gain Control is enabled on the signal from this local input.

**Valuespace:**
<On/Off>

**Example:**
```
xStatus Audio Input LocalInput 46 AGC
```
```
*s Audio Input LocalInput 46 AGC: "On"
** end
```

The Audio status, cont...

**xStatus Audio Input LocalInput [1..n] Connector**
States the connectors mixed together in the local input with the given input ID.

**Valuespace:**
<Microphone/Line/HDMI.1..8>

**Example:**
```
xStatus Audio Input LocalInput 46 Connector
```
```
*s Audio Input LocalInput 46 Connector: "Microphone.1"
** end
```

**xStatus Audio Input RemoteInput [1..n] CallId**
States which CallId the remote input with the given input ID belongs to.

**Valuespace:**
<Integer value>

**Example:**
```
xStatus Audio Input RemoteInput 51 CallId
```
```
*s Audio Input RemoteInput 51 CallId: 28
** end
```

**xStatus Audio Input RemoteInput [1..n] AGC**
States whether Automatic Gain Control is enabled on the signal from this remote input.

**Valuespace:**
<On/Off>

**Example:**
```
xStatus Audio Input RemoteInput 51 AGC
```
```
*s Audio Input RemoteInput 51 AGC: "Off"
** end
```
The Audio status, cont...

**xStatus Audio Output LocalOutput [1..n] Name**

States the name of the local output with the given output ID.

**Valuespace:**

<S: 0..255>

**Example:**

```plaintext
xStatus Audio Output LocalOutput 48 Name
```

```
*s Audio Output LocalOutput 48 Name: "MyLocalOutput1"
** end
```

**xStatus Audio Output LocalOutput [1..n] Loudspeaker**

States whether Loudspeaker is enabled on the local output with the given output ID. If one or more of the output connectors attached to this local output are connected to a loudspeaker, this signal should be a reference signal to the echo canceller and Loudspeaker should be set to On.

**Valuespace:**

<On/Off>

**Example:**

```plaintext
xStatus Audio Output LocalOutput 48 Loudspeaker
```

```
*s Audio Output LocalOutput 48 Loudspeaker: "Off"
** end
```

**xStatus Audio Output LocalOutput [1..n] Channels**

States whether the local and remote inputs connected to this local output are mixed into a mono signal or a stereo signal.

**Valuespace:**

<1..2>

**Example:**

```plaintext
xStatus Audio Output LocalOutput 48 Channels
```

```
*s Audio Output LocalOutput 48 Channels: "1"
** end
```

**xStatus Audio Output LocalOutput [1..n] Connector**

Lists the connectors attached to the local output with the given output ID.

**Valuespace:**

<Line/HDMI.1..6>

**Example:**

```plaintext
xStatus Audio Output LocalOutput 48 Connector
```

```
*s Audio Output LocalOutput 48 Connector: "Line.1"
** end
```

**xStatus Audio Output LocalOutput [1..n] Input [1..n] Gain**

Shows the gain in dB used on the input with the given input ID if this is connected to the local output with the given output ID.

**Valuespace:**

<-54..15>

**Example:**

```plaintext
xStatus Audio Output LocalOutput 48 Input 51 Gain
```

```
*s Audio Output LocalOutput 48 Input 51 Gain: 0
** end
```

**xStatus Audio Output RemoteOutput [1..n] CallId**

States which CallId the remote output with the given output ID belongs to.

**Valuespace:**

<Integer value>

**Example:**

```plaintext
xStatus Audio Output RemoteOutput 52 CallId
```

```
*s Audio Output RemoteOutput 52 CallId: 28
** end
```
The Audio status, cont...

**xStatus Audio Output RemoteOutput [1..n] Input [1..n] Gain**

Shows the gain in dB used on the input with the given input ID if this is connected to the remote output with the given output ID.

*Valuespace:*

<-54..15>

*Example:*

```c
xStatus Audio Output RemoteOutput 52 Input 46 Gain
*s Audio Output RemoteOutput 52 Input 46 Gain: 0
** end
```

**xStatus Audio Module [1..n] Type**

States the type of the audio module with the given ID.

*Valuespace:*

<DigitalNAM/Unknown>

*Example:*

```c
xStatus Audio Module 1 Type
*s Audio Module 1 Type: DigitalNAM
** end
```

**xStatus Audio Module [1..n] SoftwareID**

States the SoftwareID of the DNAM dsp software.

*Valuespace:*

<Integer value>

*Example:*

```c
xStatus Audio Module 1 SoftwareID
*s Audio Module 1 SoftwareID: “114”
** end
```

The Audio status, cont...

**xStatus Audio Module [1..n] HardwareID**

States the DNAM HardwareID.

*Valuespace:*

<S: 0..255>

*Example:*

```c
xStatus Audio Module 1 HardwareID
*s Audio Module 1 HardwareID: “B40F69”
** end
```

**xStatus Audio Module [1..n] Connector**

States which output connector the audio module with the given ID is attached to.

*Valuespace:*

<Line_out.1/3>

*Example:*

```c
xStatus Audio Module 1 Connector
*s Audio Module 1 Connector: “Line_out.1”
** end
```
The Call status

**xStatus Call**
Run this command to see an overview of the call status. The call id can be used later 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 RemoteNumber: "john.doe@company.com"
*s Call 27 CallbackNumber: "sip:john.doe@company.com"
*s Call 27 DisplayName: "john.doe"
*s Call 27 CallRate: 968
*s Call 27 TransmitCallRate: 768
*s Call 27 ReceiveCallRate: 968
*s Call 27 Encryption Type: "None"
*s Call 27 PlacedOnHold: False
** end
```

**xStatus Call [1..n] Status**
Status of a call.

Valuespace:
- Dialling/Connecting/Ringing/Connected/Idle

Example:
```
xStatus Call 27 Status
*s Call 28 Status: Connected
** end
```

**xStatus Call [1..n] Direction**
Direction of the call initiation.

Valuespace:
- Incoming/Outgoing

Example:
```
xStatus Call 28 Direction
*s Call 28 Direction: Outgoing
** end
```

**xStatus Call [1..n] Protocol**
Call protocol, H323 or SIP.

Valuespace:
- H323/SIP

Example:
```
xStatus Call 28 Protocol
*s Call 28 Protocol: "h323"
** end
```

**xStatus Call [1..n] RemoteNumber**
The number or URI dialled to the far end site.

Valuespace:
- S: 0..100

Example:
```
xStatus Call 28 RemoteNumber
*s Call 28 RemoteNumber: "5585232"
** end
```

**xStatus Call [1..n] CallbackNumber**
Far end site dial number. Includes call protocol.

Valuespace:
- S: 0..100

Example:
```
xStatus Call 28 CallbackNumber
*s Call 28 CallbackNumber: "h323:john.doe@company.com"
** end
```
The Call status, cont...

**xStatus Call [1..n] Display Name**
Name to display for far end site.

Valuespace:
*S: 0,100>

Example:
```
xStatus Call 28 DisplayName
*s Call 28 DisplayName: “john.doe@company.com”
** end
```

**xStatus Call [1..n] Call Rate**
Call bandwidth in kilobits per second.

Valuespace:
<Integer value>

Example:
```
xStatus Call 28 CallRate
*s Call 28 CallRate: 968
** end
```

**xStatus Call [1..n] Transmit Call Rate**
Transmitted bandwidth in the call in kilobits per second.

Valuespace:
<Integer value>

Example:
```
xStatus Call 28 TransmitCallRate
*s Call 28 TransmitCallRate: 768
** end
```

**xStatus Call [1..n] Receive Call Rate**
Received bandwidth in the call in kilobits per second.

Valuespace:
<Integer value>

Example:
```
xStatus Call 28 ReceiveCallRate
*s Call 28 ReceiveCallRate: 968
** end
```

**xStatus Call [1..n] Encryption Type**
Encryption standard.

Valuespace:
=None/Aes-128>

Example:
```
xStatus Call 28 Encryption Type
*s Call 28 Encryption Type: “None”
** end
```

**xStatus Call [1..n] Placed On Hold**
This status tells whether the call is placed on hold or not.

Valuespace:
=True/False>

Example:
```
xStatus Call 28 PlacedOnHold
*s Call 28 PlacedOnHold: False
** end
```
The Camera status

**xStatus Camera**

Run this command to see an overview of the camera status.

**Example:**

```plaintext
xStatus Camera

*s Camera 1 Connected: True
*s Camera 1 HardwareID: "50000000"
*s Camera 1 Manufacturer: "TANDBERG"
*s Camera 1 Model: "PrecisionHD 1080p 12X"
*s Camera 1 SoftwareID: "S01718-4.0FINAL [ID:40043] 2010-01-22"
*s Camera 1 IpAddress: "-"
*s Camera 1 MacAddress: "-"
*s Camera 1 Position Pan: 440
*s Camera 1 Position Tilt: 69
*s Camera 1 Position Zoom: 1
*s Camera 1 Position Focus: 4581

- continues with status for Camera 2-7
** end
```

**xStatus Camera [1..7] Connected**

Indicates whether the given camera is connected or not.

**Valuespace:**

<True/False>

**Example:**

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

**xStatus Camera [1..7] HardwareID**

States the hardware id of the camera.

**Valuespace:**

<S: 0..100>

**Example:**

```plaintext
xStatus Camera 1 HardwareID
*s Camera 1 HardwareID: "50000000"
** end
```

**xStatus Camera [1..7] Manufacturer**

States the manufacturer of the camera.

**Valuespace:**

<S: 0..100>

**Example:**

```plaintext
xStatus Camera 1 Manufacturer
*s Camera 1 Manufacturer: "TANDBERG"
** end
```

**xStatus Camera [1..7] Model**

States the camera model.

**Valuespace:**

<S: 0..100>

**Example:**

```plaintext
xStatus Camera 1 Model
*s Camera 1 Model: "PrecisionHD 1080p 12X"
** end
```
The Camera status, cont...

**xStatus Camera [1..7] SoftwareID**
States the software id of the camera.

Valuespace:  
<S: 0,100>

Example:  
```
xStatus Camera 1 SoftwareID
*s Camera 1 SoftwareID: "S01718-4.0FINAL [ID:40043] 2009-06-25"
** end
```

**xStatus Camera [1..7] IpAddress**
Shows the camera IP address retrieval status. See also "xCommand Cameras Camera[1..x] DHCP" command.

Valuespace:  
<S: 0,100>

Examples:  
```
xStatus Camera 1 IpAddress
*s Camera 1 IpAddress: "10.47.19.19"
** end
```

**xStatus Camera [1..7] MacAddress**
Shows the camera MacAddress retrieval status. See also "xCommand Cameras Camera[1..x] DHCP" command.

Valuespace:  
<S: 0,100>

Examples:  
```
xStatus Camera 1 MacAddress
*s Camera 1 MacAddress: "10.47.19.19"
** end
```

The Camera status, cont...

**xStatus Camera [1..7] Position Pan**
Current pan position. Value range depends on camera type.

Valuespace:  
<-65535..65535>

Example:  
```
xStatus Camera 1 Position Pan
*s Camera 1 Position Pan: 514
** end
```

**xStatus Camera [1..7] Position Tilt**
Current tilt position. Value range depends on camera type.

Valuespace:  
<-65535..65535>

Example:  
```
xStatus Camera 1 Position Tilt
*s Camera 1 Position Tilt: 142
** end
```

**xStatus Camera [1..7] Position Zoom**
Current zoom position. Value range depends on camera type.

Valuespace:  
<0..65535>

Example:  
```
xStatus Camera 1 Position Zoom
*s Camera 1 Position Zoom: 1636
** end
```
The Camera status, cont...

**xStatus Camera [1..7] Position Focus**
Current focus position. Value range will depend on camera type.

**Valuespace:**
<0..65535>

**Example:**
```
xStatus Camera 1 Position Focus
*s Camera 1 Position Focus: 4474
** end
```

The Conference status

**xStatus Conference**
Run this command to see an overview of the conference status.

**Example:**
```
xStatus Conference
*s Conference Presentation Mode: Off
*s Conference Presentation Protocol: 
*s Conference Presentation Resolution Height: 0
*s Conference Presentation Resolution Width: 0
*s Conference Presentation SiteId: 0
*s Conference Presentation LocalSource: 0
*s Conference Site 27 MicrophonesMuted: True
** end
```

**xStatus Conference Presentation Mode**
Secondary video stream status.

**Valuespace:**
<Off/Sending/Receiving>

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

**xStatus Conference Presentation Protocol**
Video protocol used for transmitting the presentation

**Valuespace:**
<0..10>

**Example:**
```
xStatus Conference Presentation Protocol
*s Conference Presentation Protocol: "H264"
** end
```
The Conference status, cont...

**xStatus Conference Presentation Resolution Height**
Current resolution height for presentation.

Valuespace:

<0..3000>

**Example:**

```c
*xStatus Conference Presentation Resolution Height
*s Conference Presentation Resolution Height: 0
** end
```

**xStatus Conference Presentation Resolution Width**
Current resolution width for presentation.

Valuespace:

<0..4000>

**Example:**

```c
*xStatus Conference Presentation Resolution Width
*s Conference Presentation Resolution Width: 0
** end
```

**xStatus Conference Presentation SiteId**
Id of site currently sending presentation.

Valuespace:

<0..65535>

**Example:**

```c
*xStatus Conference Presentation SiteId
*s Conference Presentation SiteId: 0
** end
```

**xStatus Conference Presentation LocalSource**
Local video source that will be used when presentation is sent from local site.

Valuespace:

<1..5>

**Example:**

```c
*xStatus Conference Presentation LocalSource
*s Conference Presentation LocalSource: 0
** end
```

**xStatus Conference Site [1..n] MicrophonesMuted**
Lists audio mute status for other sites participating in conference with the given site id.

Valuespace:

<True/False>

**Example:**

```c
*xStatus Conference Site 27 MicrophonesMuted
*s Conference Site 27 MicrophonesMuted: True
** end
```
The Diagnostics status

xStatus Diagnostics

Run this command to see an overview of the diagnostics. The example shows the diagnostics status for an ongoing call. The call id and the id’s for incoming/outgoing audio, video and data channels are required when query diagnostics 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 MaxJitter: 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 MaxJitter: 9
*s Diagnostics Call 27 Channels IncomingVideoChannel 333 Netstat 1 Jitter: 0
*s Diagnostics Call 27 Channels IncomingVideoChannel 333 Netstat 1 Packets: 0
*s Diagnostics Call 27 Channels IncomingVideoChannel 333 Netstat 1 Loss: 0
*s Diagnostics Call 27 Channels IncomingVideoChannel 333 Netstat 1 MaxJitter: 0
*s Diagnostics Call 27 Channels IncomingVideoChannel 335 Netstat 1 Jitter: 0
*s Diagnostics Call 27 Channels IncomingVideoChannel 335 Netstat 1 Packets: 0
*s Diagnostics Call 27 Channels IncomingVideoChannel 335 Netstat 1 Loss: 0
*s Diagnostics Call 27 Channels IncomingVideoChannel 335 Netstat 1 MaxJitter: 0
*s Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 Drop: 0
*s Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 Bytes: 0
*s Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 ChannelRate: 0
*s Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 MaxJitter: 0
*s Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 Jitter: 0
*s Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 Packets: 123043
*s Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 Loss: 0
*s Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 Jitter: 4
*s Diagnostics Call 27 Channels OutgoingVideoChannel 334 Netstat 1 Jitter: 4
*s Diagnostics Call 27 Channels OutgoingVideoChannel 334 Netstat 1 Packets: 0
*s Diagnostics Call 27 Channels OutgoingVideoChannel 334 Netstat 1 Loss: 0
*s Diagnostics Call 27 Channels OutgoingVideoChannel 336 Netstat 1 Jitter: 0
*s Diagnostics Call 27 Channels OutgoingVideoChannel 336 Netstat 1 Packets: 0
*s Diagnostics Call 27 Channels OutgoingVideoChannel 336 Netstat 1 Loss: 0
*s Diagnostics Call 27 Channels OutgoingVideoChannel 336 Netstat 1 ChannelRate: 0
*s Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 Jitter: 0
*s Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 Packets: 0
*s Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 Loss: 0
*s Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 ChannelRate: 0
** end
The Diagnostics status, cont...

**xStatus** Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1

**Jitter**

Current jitter as specified by RFC3550.

**Valuespace:**

<Integer value>

**Example:**

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

---

**xStatus** Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1

**Packets**

Number of packets received in this media channel.

**Valuespace:**

<Integer value>

**Example:**

```
xStatus Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 Packets
  /* Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 Packets: 405 */
** end
```

---

**xStatus** Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1

**Loss**

Packets lost in this media channel.

**Valuespace:**

<Integer value>

**Example:**

```
xStatus Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 Loss
  /* Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 Loss: 96 */
** end
```

---

**xStatus** Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1

**Bytes**

Number of bytes received in this media channel.

**Valuespace:**

<Integer value>

**Example:**

```
xStatus Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 Bytes
  /* Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 Bytes: 129920 */
** end
```

---

**xStatus** Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1

**ChannelRate**

Current bandwidth for this media channel.

**Valuespace:**

<Integer value>

**Example:**

```
xStatus Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 ChannelRate
  /* Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 ChannelRate: 128000 */
** end
```
The Diagnostics status, cont...

**xStatus Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1 MaxJitter**
Max jitter measured during last time interval (5 seconds).

**Valuespace:**
<Integer value>

**Example:**
```
xStatus Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 MaxJitter
*s Diagnostics Call 27 Channels IncomingAudioChannel 327 Netstat 1 MaxJitter: 0
** end
```

**xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 Jitter**
Current jitter as specified by RFC3550.

**Valuespace:**
<Integer value>

**Example:**
```
xStatus Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 Jitter
*s Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 Jitter: 6
** end
```

**xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 Packets**
Number of packets received in this media channel.

**Valuespace:**
<Integer value>

**Example:**
```
xStatus Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 Packets
*s Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 Packets: 38699
** end
```

The Diagnostics status, cont...

**xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 Loss**
Packets lost in this media channel.

**Valuespace:**
<Integer value>

**Example:**
```
xStatus Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 Loss
*s Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 Loss: 0
** end
```

**xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 Drop**
Packets dropped in this media channel.

**Valuespace:**
<Integer value>

**Example:**
```
xStatus Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 Drop
*s Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 Drop: 0
** end
```

**xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 Bytes**
Number of bytes received in this media channel.

**Valuespace:**
<Integer value>

**Example:**
```
xStatus Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 Bytes
*s Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 Bytes: 0
** end
```
The Diagnostics status, cont...

xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 ChannelRate

Current bandwidth for this media channel.

Valuespace:
<Integer value>

Example:

xStatus Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 ChannelRate
*s Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 ChannelRate: 0
** end

xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 MaxJitter

Max jitter measured during last time interval (5 seconds).

Valuespace:
<Integer value>

Example:

xStatus Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 MaxJitter
*s Diagnostics Call 27 Channels IncomingVideoChannel 330 Netstat 1 MaxJitter: 0
** end

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 Jitter

Current jitter as specified by RFC3550.

Valuespace:
<Integer value>

Example:

xStatus Diagnostics Call 27 Channels IncomingDataChannel 335 Netstat 1 Jitter
*s Diagnostics Call 27 Channels IncomingDataChannel 335 Netstat 1 Jitter: 0
** end

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 Packets

Number of packets received in this media channel.

Valuespace:
<Integer value>

Example:

xStatus Diagnostics Call 27 Channels IncomingDataChannel 335 Netstat 1 Packets
*s Diagnostics Call 27 Channels IncomingDataChannel 335 Netstat 1 Packets: 10
** end

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 Loss

Packets lost in this media channel.

Valuespace:
<Integer value>

Example:

xStatus Diagnostics Call 27 Channels IncomingDataChannel 335 Netstat 1 Loss
*s Diagnostics Call 27 Channels IncomingDataChannel 335 Netstat 1 Loss: 0
** end

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 Drop

Packets dropped in this media channel.

Valuespace:
<Integer value>

Example:

xStatus Diagnostics Call 27 Channels IncomingDataChannel 335 Netstat 1 Drop
*s Diagnostics Call 27 Channels IncomingDataChannel 335 Netstat 1 Drop: 0
** end
The Diagnostics status, cont...

**xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 Bytes**

Number of bytes received in this media channel.

**Valuespace:**

<Integer value>

**Example:**

```plaintext
xStatus Diagnostics Call 27 Channels IncomingDataChannel 335 Netstat 1 Bytes
```

**xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 ChannelRate**

Current bandwidth for this media channel.

**Valuespace:**

<Integer value>

**Example:**

```plaintext
xStatus Diagnostics Call 27 Channels IncomingDataChannel 335 Netstat 1 ChannelRate
```

**xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 MaxJitter**

Max jitter measured during last time interval (5 seconds).

**Valuespace:**

<Integer value>

**Example:**

```plaintext
xStatus Diagnostics Call 27 Channels IncomingDataChannel 335 Netstat 1 MaxJitter
```

The Diagnostics status, cont...

**xStatus Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 Jitter**

Current jitter as specified by RFC3550.

**Valuespace:**

<Integer value>

**Example:**

```plaintext
xStatus Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 Jitter
```

**xStatus Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 Packets**

Number of packets received in this media channel.

**Valuespace:**

<Integer value>

**Example:**

```plaintext
xStatus Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 Packets
```

**xStatus Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 Loss**

Packets lost in this media channel.

**Valuespace:**

<Integer value>

**Example:**

```plaintext
xStatus Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 Loss
```
The Diagnostics status, cont...

**xStatus**

**Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 Drop**

Packets dropped in this media channel.

**Valuespace:**

<Integer value>

**Example:**

```plaintext
xStatus Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 Drop
*s Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 Drop: 0
** end
```

**xStatus**

**Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 Bytes**

Number of bytes sent in this media channel.

**Valuespace:**

<Integer value>

**Example:**

```plaintext
xStatus Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 Bytes
*s Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 Bytes: 7653930
** end
```

**xStatus**

**Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 ChannelRate**

Current bandwidth for this media channel.

**Valuespace:**

<Integer value>

**Example:**

```plaintext
xStatus Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 ChannelRate
*s Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 ChannelRate: 127000
** end
```

**xStatus**

**Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 MaxJitter**

Max jitter measured during last time interval (5 seconds).

**Valuespace:**

<Integer value>

**Example:**

```plaintext
xStatus Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 MaxJitter
*s Diagnostics Call 27 Channels OutgoingAudioChannel 328 Netstat 1 MaxJitter: 0
** end
```

**xStatus**

**Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 Jitter**

Current jitter as specified by RFC3550.

**Valuespace:**

<Integer value>

**Example:**

```plaintext
xStatus Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 Jitter
*s Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 Jitter: 1
** end
```

**xStatus**

**Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 Packets**

Number of packets received in this media channel.

**Valuespace:**

<Integer value>

**Example:**

```plaintext
xStatus Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 Packets
*s Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 Packets: 43096
** end
```
The Diagnostics status, cont...

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 Loss
   Packets lost in this media channel.

   Valuespace:
      <Integer value>

   Example:
      xStatus Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 Loss
      *s Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 Loss: 0
      ** end

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 Drop
   Packets dropped in this media channel.

   Valuespace:
      <Integer value>

   Example:
      xStatus Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 Drop
      *s Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 Drop: 0
      ** end

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 Bytes
   Number of bytes sent in this media channel.

   Valuespace:
      <Integer value>

   Example:
      xStatus Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 Bytes
      *s Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 Bytes: 39111160
      ** end

The Diagnostics status, cont...

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 ChannelRate
   Current bandwidth for this media channel.

   Valuespace:
      <Integer value>

   Example:
      xStatus Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 ChannelRate
      *s Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 ChannelRate: 652000
      ** end

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 MaxJitter
   Max jitter measured during last time interval (5 seconds).

   Valuespace:
      <Integer value>

   Example:
      xStatus Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 MaxJitter
      *s Diagnostics Call 27 Channels OutgoingVideoChannel 331 Netstat 1 MaxJitter: 1
      ** end

xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 Jitter
   Current jitter as specified by RFC3550.

   Valuespace:
      <Integer value>

   Example:
      xStatus Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 Jitter
      *s Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 Jitter: 0
      ** end
The Diagnostics status, cont...

**xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 Packets**
Number of packets received in this media channel.

**Valuespace:**

\(<\text{Integer value}>\)

**Example:**

```c
xStatus Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 Packets
*s Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 Packets: 2
** end
```

**xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 Loss**
Packets lost in this media channel.

**Valuespace:**

\(<\text{Integer value}>\)

**Example:**

```c
xStatus Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 Loss
*s Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 Loss: 0
** end
```

**xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 Drop**
Packets dropped in this media channel.

**Valuespace:**

\(<\text{Integer value}>\)

**Example:**

```c
xStatus Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 Drop
*s Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 Drop: 0
** end
```

**xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 ChannelRate**
Current bandwidth for this media channel.

**Valuespace:**

\(<\text{Integer value}>\)

**Example:**

```c
xStatus Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 ChannelRate
*s Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 ChannelRate: 0
** end
```

**xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 MaxJitter**
Max jitter measured during last time interval (5 seconds).

**Valuespace:**

\(<\text{Integer value}>\)

**Example:**

```c
xStatus Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 MaxJitter
*s Diagnostics Call 27 Channels OutgoingDataChannel 336 Netstat 1 MaxJitter: 0
** end
```
The GPIO status

**xStatus GPIO Pin [1..4] State**

NOTE: This command is not supported on Codec C40.

Shows the current state of each GPIO pin. The default state is High (+12V). The state is Low (0V) when activated.

Example:
```
xStatus GPIO Pin 1 State
*s GPIO Pin 1 State: High
** end
```

The H323 status

**xStatus H323**

Run this command to see an overview of the H323 status.

Example:
```
xStatus H323
*s H323 Gatekeeper Status: Registered
*s H323 Gatekeeper Address: "10.47.1.58"
*s H323 Gatekeeper Port: 1719
*s H323 Gatekeeper Reason: ""
** end
```

**xStatus H323 Gatekeeper Status**
Gatekeeper registration status.

Valuespace:
<Registered/Inactive/Rejected>

Example:
```
xStatus H323 Gatekeeper Status
*s H323 Gatekeeper Status: Registered
** end
```

**xStatus H323 Gatekeeper Address**
Address of gatekeeper this system is registered against.

Valuespace:
<S: 0..100>

Example:
```
xStatus H323 Gatekeeper Address
*s H323 Gatekeeper Address: "10.47.1.83"
** end
```
The H323 status, cont...

xStatus H323 Gatekeeper Port
The port to connect to on gatekeeper.

Valuespace:
<Integer value>

Example:
xStatus H323 Gatekeeper Port
*s H323 Gatekeeper Port: 1719
** end

xStatus H323 Gatekeeper Reason
Reason for rejected registration.

Valuespace:
<S: 0,100>

Example:
xStatus H323 Gatekeeper Reason
*s H323 Gatekeeper Reason: ""
** end

The HTTP Feedback status

xStatus HTTPFeedb...
The HTTP Feedback status, cont...

**xStatus HttpFeedback [1..3] Expression**

A set of 15 feedback expressions can be registered for each URL.

**Valuespace:**

&S: 0..256

**Example:**

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 Media Channel status

**xStatus MediaChannels**

Run this command to see an overview of the media channel status. The example shows the media channel status for an ongoing call. The call id and the id's for incoming/outgoing audio and video channels are required when query media channels information.

**Example:**

xStatus Media

*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 Mute: True
*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 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 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 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 ResolutionX: 0
*s MediaChannels Call 27 OutgoingVideoChannel 334 Video ResolutionY: 0
** end
The Media Channel status, cont...

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Encryption Status
Encryption status on media channel.

Valuespace:
<On/off>

Example:
```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Encryption Status
  *x MediaChannels Call 27 IncomingAudioChannel 327 Encryption Status: Off
** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Audio Protocol
Audio algorithm.

Valuespace:
<AACLD/G722/G7221/G711Mu>

Example:
```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Audio Protocol
  *x MediaChannels Call 27 IncomingAudioChannel 327 Audio Protocol: AACLD
** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Audio Mute
Audio mute status of incoming audio.

Valuespace:
<True/False>

Example:
```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Audio Mute
  *x MediaChannels Call 27 IncomingAudioChannel 327 Audio Mute: True
** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Audio Channels
Number of audio channels.

Valuespace:
<Integer value>

Example:
```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Audio Channels
  *x MediaChannels Call 27 IncomingAudioChannel 327 Audio Channels: 1
** end
```

The Media Channel status, cont...

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Encryption Status
Encryption status on media channel.

Valuespace:
<On/Off>

Example:
```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Encryption Status
  *x MediaChannels Call 27 IncomingVideoChannel 330 Encryption Status: Off
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] ChannelRole
Main video channel or presentation video channel.

Valuespace:
>Main/Presentation>

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

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Video Protocol
Video algorithm.

Valuespace:
<H264/H263pp/H263/H261>

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

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Video ResolutionX
Resolution width for incoming video.

Valuespace:
<Integer value>

Example:
```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Video ResolutionX
  *x MediaChannels Call 27 IncomingVideoChannel 330 Video ResolutionX: 768
** end
```
The Media Channel status, cont...

**xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Video ResolutionY**
Resolution height for incoming video.

**Valuespace:**
<Integer value>

**Example:**
- xStatus MediaChannels Call 27 IncomingVideoChannel 330 Video ResolutionY
  - MediaChannels Call 27 IncomingVideoChannel 330 Video ResolutionY: 448
** end

**xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Encryption Status**
Encryption status on media channel.

**Valuespace:**
<On/Off>

**Example:**
- xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Encryption Status
  - MediaChannels Call 27 OutgoingAudioChannel 328 Encryption Status: Off
** end

**xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Audio Protocol**
Audio algorithm.

**Valuespace:**
<AACLD/G722/G7221/G711Mu>

**Example:**
- xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Audio Protocol
  - MediaChannels Call 27 OutgoingAudioChannel 328 Audio Protocol: AACLD
** end

**xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Encryption Status**
Encryption status on media channel.

**Valuespace:**
<On/Off>

**Example:**
- 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**
Main video channel or presentation video channel.

**Valuespace:**
<Main/Presentation>

**Example:**
- xStatus MediaChannels Call 27 OutgoingVideoChannel 331 ChannelRole
  - MediaChannels Call 27 OutgoingVideoChannel 331 ChannelRole: Main
** end
The Media Channel status, cont...

**xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Video Protocol**

Video algorithm.

*Valuespace:*

    `<H264/H263pp/H263/H261>`

*Example:*

    ```
    xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video Protocol
    `s MediaChannels Call 27 OutgoingVideoChannel 331 Video Protocol: "H264"
    ** end
    ```

**xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Video ResolutionX**

Resolution width for incoming video.

*Valuespace:*

    `<Integer value.>`

*Example:*

    ```
    xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video ResolutionX
    `s MediaChannels Call 27 OutgoingVideoChannel 331 Video ResolutionX: 768
    ** end
    ```

**xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Video ResolutionY**

Resolution height for incoming video.

*Valuespace:*

    `<Integer value.>`

*Example:*

    ```
    xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video ResolutionY
    `s MediaChannels Call 27 OutgoingVideoChannel 331 Video ResolutionY: 448
    ** end
    ```

The Network status

**xStatus Network**

Run this command to see an overview of the network status.

*Example:*

    ```
    xStatus Network
    `s Network 1 Ethernet MacAddress: "00:50:60:02:E7:D3"
    `s Network 1 Ethernet Speed: "1000full"
    `s Network 1 IPv4 Address: "10.47.21.62"
    `s Network 1 IPv4 SubnetMask: "255.255.255.0"
    `s Network 1 IPv4 Gateway: "10.47.21.1"
    `s Network 1 IPv4 DNS Domain Name: "sales.company.com eu.company.int"
    `s Network 1 IPv4 DNS Server 1 Address: "10.47.21.61"
    `s Network 1 IPv4 DNS Server 2 Address: "10.47.21.14"
    `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 IPv4 MTU: 1500
    ** end
    ```

**xStatus Network 1 Ethernet MacAddress**

Mac address for ethernet interface.

*Valuespace:*

    `<S: 0,.100>`

*Example:*

    ```
    xStatus Network 1 Ethernet MacAddress
    `s Network 1 Ethernet MacAddress: "00:50:60:02:FD:C7"
    ** end
    ```

**xStatus Network 1 Ethernet Speed**

Speed in Mbps. Full or half duplex.

*Valuespace:*

    `<Integer value>`

*Example:*

    ```
    xStatus Network 1 Ethernet Speed
    `s Network 1 Ethernet Speed: "100full"
    ** end
    ```
The Network status, cont...

**xStatus Network 1 IPv4 Address**
Ip address of this system.

*Valuespace:*
\(<S: 0,.100>\)

*Example:*

```plaintext
xStatus Network 1 IPv4 Address
*s Network 1 IPv4 Address: "10.47.21.62"
** end
```

**xStatus Network 1 IPv4 SubnetMask**
A mask used to determine what subnet the IPv4 address belongs to.

*Valuespace:*
\(<S: 0,.100>\)

*Example:*

```plaintext
xStatus Network 1 IPv4 SubnetMask
*s Network 1 IPv4 SubnetMask: "255.255.255.0"
** end
```

**xStatus Network 1 IPv4 Gateway**
The gateway used.

*Valuespace:*
\(<S: 0,.100>\)

*Example:*

```plaintext
xStatus Network 1 IPv4 Gateway
*s Network 1 IPv4 Gateway: "10.47.21.1"
** end
```

**xStatus Network 1 IPv4 DNS Domain Name**
The Domain Name

*Valuespace:*
\(<S: 0,.100>\)

*Example:*

```plaintext
xStatus Network 1 IPv4 DNS Domain Name
*s Network 1 IPv4 DNS Domain Name: "sales.company.com emea.company.int"
** end
```

**xStatus Network 1 IPv4 DNS Server [1..5] Address**
The IP address of the DNS server to use.

*Valuespace:*
\(<S: 0,.100>\)

*Example:*

```plaintext
xStatus Network 1 IPv4 DNS Server 1 Address
*s Network 1 IPv4 DNS Server 1 Address: "10.47.1.92"
** end
```

**xStatus Network 1 IPv4 MTU**
Specifies the MTU (Maximum Transmission Unit) size for the network.

*Valuespace:*
\(<\text{Integer value}>\)

*Example:*

```plaintext
xStatus Network 1 IPv4 MTU
*s Network 1 IPv4 MTU: 1500
** end
```
The Preset status

**xStatus Preset**
Run this command to see an 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.

**Valuespace:**
<True/False>

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

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

**Valuespace:**
<All/...>

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

xStatus SIP
Run this command to see an overview of the SIP status.

Example:
```
xStatus SIP
*s SIP Proxy 1 Status: Active
*s SIP Proxy 1 Address: "10.47.21.58"
*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: "john.doe@company.com"
*s SIP Registration 1 Authentication: Off
*s SIP Profile 1 Proxy 1 Status: Active
*s SIP Profile 1 Proxy 1 Address: "10.47.1.58"
*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"
** end
```

The SIP status, cont...

xStatus SIP Proxy 1 Address
The active proxy this system communicates with.

Valuespace:
```
<S: 0..255>
```

Example:
```
xStatus SIP Proxy 1 Address
*s SIP Proxy 1 Address: "10.47.1.63"
** end
```

xStatus SIP Proxy 1 Secure
Encryption status of signalling with proxy.

Valuespace:
```
<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.

Valuespace:
```
<True/False>
```

Example:
```
xStatus SIP Proxy 1 Verified
*s SIP Proxy 1 Verified: False
** end
```

The SIP status

xStatus SIP Proxy 1 Status
Status of communication between this endpoint and SIP proxy.

Valuespace:
```
<Active/DNSFailed/Off/Timeout/UnableTCP/UnableTLS/Unknown>
```

Example:
```
xStatus SIP Proxy 1 Status
*s SIP Proxy 1 Status: Active
** end
```
The SIP status, cont...

**xStatus SIP Registration [1..n] Status**
Registration status of SIP Registration URI.

Valuespace:
- <Deregister/Failed/Inactive/Registered/Registering>

Example:
```
xStatus SIP Registration 1 Status
*s SIP Registration 1 Status: Registered
** end
```

**xStatus SIP Registration [1..n] Reason**
Specifies the reason why a SIP registration fails.

Valuespace:
- <S: 0..100>

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

**xStatus SIP Registration [1..n] URI**
The URI used for registration vs SIP registrar.

Valuespace:
- <S: 0..100>

Example:
```
xStatus SIP Registration 1 URI
*s SIP Registration 1 URI: "john.doe@company.com"
** end
```

The SIP status, cont...

**xStatus SIP Registration [1..n] Authentication**
States what Authentication mechanism is used.

Valuespace:
- <Digest/NTLM/Off>

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

**xStatus SIP Profile 1 Proxy 1 Status**
Status of communication between this endpoint and SIP proxy, for the SIP Profile 1.

Valuespace:
- <Active/DNSFailed/Off/Timeout/UnableTCP/UnableTLS/Unknown>

Example:
```
xStatus SIP Profile 1 Proxy 1 Status
*s SIP Profile 1 Proxy 1 Status: Active
** end
```

**xStatus SIP Profile 1 Proxy 1 Address**
The active proxy this system do communicate with, for the SIP Profile 1.

Valuespace:
- <S: 0..255>

Example:
```
xStatus SIP Profile 1 Proxy 1 Address
*s SIP Profile 1 Proxy 1 Address: "10.47.1.63"
** end
```
The SIP status, cont...

**xStatus SIP Profile 1 Secure**
Encryption status of signalling with proxy, for the SIP Profile 1.

**Valuespace:**
<True/False>

**Example:**
```
xStatus SIP Profile 1 Secure
*s SIP Profile 1 Secure: True
** end
```

**xStatus SIP Profile 1 Verified**
For TLS connections a CA certificate can be uploaded from the web interface.

True: Only TLS connections to servers, whom the x.509 certificate is validated against in the CA-list, will be allowed.
False: No certificate has been uploaded, or the certificate could not be verified.

**Valuespace:**
<True/False>

**Example:**
```
xStatus SIP Profile 1 Verified
*s SIP Profile 1 Verified: False
** end
```

**xStatus SIP Profile 1 Authentication**
States what Authentication mechanism is used.

**Valuespace:**
<Digest/NTLM/Off>

**Example:**
```
xStatus SIP Profile 1 Authentication
*s SIP Profile 1 Authentication: Off
** end
```

The SIP status, cont...

**xStatus SIP Profile 1 Registration [1..n] Status**
Registration status of SIP Registration URI, for the SIP Profile 1.

**Valuespace:**
<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**
Specifies the reason why a SIP registration fails, for the SIP Profile 1.

**Valuespace:**
<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**
The URI used for registration vs SIP registrar, for the SIP Profile 1.

**Valuespace:**
<S: 0,.100>

**Example:**
```
xStatus SIP Profile 1 Registration 1 URI
*s SIP Profile 1 Registration 1 URI: "john.doe@company.com"
** end
```
The Standby status

xStatus Standby Active
States whether standby mode is active.

Valuespace:
<On/Off>

Example:

xStatus Standby Active
*s Standby Active: Off
** end

The System Unit status

xStatus SystemUnit
Run this command to see an overview of the system unit status.

Example:

xStatus SystemUnit
*s SystemUnit ProductType: "TANDBERG Codec"
*s SystemUnit ProductId: "TANDBERG Codec C60"
*s SystemUnit ProductPlatform: "C60"
*s SystemUnit Uptime: 597095
*s SystemUnit Software Application: "Endpoint"
*s SystemUnit Software Version: "TC3.0.0"
*s SystemUnit Software Name: "s52000"
*s SystemUnit Software ReleaseDate: "2010-01-26"
*s SystemUnit Software MaxVideoCalls: 3
*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 2009.03-37"
*s SystemUnit State System: Initialized
*s SystemUnit State MaxNumberOfCalls: 3
*s SystemUnit State MaxNumberOfActiveCalls: 3
*s SystemUnit State NumberOfActiveCalls: 1
*s SystemUnit State NumberOfSuspendedCalls: 0
*s SystemUnit State NumberOfInProgressCalls: 0
*s SystemUnit State Subsystem Application: Initialized
*s SystemUnit ContactInfo: "support@company.com"
** end
The System Unit status, cont...

xStatus SystemUnit ProductType
Displays the product type.

Valuespace:
<S: 0..100>

Example:

*xStatus SystemUnit ProductType
*s SystemUnit ProductType: "TANDBERG Codec"
** end

xStatus SystemUnit ProductId
Displays what product this is.

Valuespace:
<S: 0..100>

Example:

*xStatus SystemUnit ProductId
*s SystemUnit ProductId: "TANDBERG Codec C60"
** end

xStatus SystemUnit ProductPlatform
Displays the product platform.

Valuespace:
<S: 0..100>

Example:

*xStatus SystemUnit ProductPlatform
*s SystemUnit ProductPlatform: "C60"
** end

xStatus SystemUnit Uptime
Seconds since last system boot.

Valuespace:
<Integer value>

Example:

*xStatus SystemUnit Uptime
*s SystemUnit Uptime: 357708
** end

The System Unit status, cont...

xStatus SystemUnit Software Application
Specifies which software application is running.

Valuespace:
<S: 0..100>

Example:

*xStatus SystemUnit Software Application
*s SystemUnit Software Application: "Endpoint"
** end

xStatus SystemUnit Software Version
Version of installed software.

Valuespace:
<S: 0..100>

Example:

*xStatus SystemUnit Software Version
*s SystemUnit Software Version: "TC3.0.0"
** end

xStatus SystemUnit Software Name
Specifies the name of the installed system.

Valuespace:
<S: 0..100>

Example:

*xStatus SystemUnit Software Name
*s SystemUnit Software Name: "s52000"
** end

xStatus SystemUnit Software ReleaseDate
Date of software release.

Valuespace:
<S: 0..100>

Example:

*xStatus SystemUnit Software ReleaseDate
*s SystemUnit Software ReleaseDate: "2009-07-03"
** end
The System Unit status, cont...

**xStatus SystemUnit Software MaxVideoCalls**
Max number of simultaneous video calls from this system.

**Valuespace:**
<Integer value>

**Example:**
```
xStatus SystemUnit Software MaxVideoCalls
!*s SystemUnit Software MaxVideoCalls: 3
** end
```

**xStatus SystemUnit Software ReleaseKey**
States whether system has a valid releasekey for this software version.

**Valuespace:**
<S: 0..100>

**Example:**
```
xStatus SystemUnit Software ReleaseKey
!*s SystemUnit Software ReleaseKey: "true"
** end
```

**xStatus SystemUnit Software OptionKeys NaturalPresenter**
States whether system supports NaturalPresenter functionality.

**Valuespace:**
<S: 0..100>

**Example:**
```
xStatus SystemUnit Software OptionKeys NaturalPresenter
!*s SystemUnit Software OptionKeys NaturalPresenter: "true"
** end
```

**xStatus SystemUnit Software OptionKeys MultiSite**
States whether system supports MultiSite functionality.

**Valuespace:**
<S: 0..100>

**Example:**
```
xStatus SystemUnit Software OptionKeys MultiSite
!*s SystemUnit Software OptionKeys MultiSite: "true"
** end
```

The System Unit status, cont...

**xStatus SystemUnit Software OptionKeys PremiumResolution**
States whether system supports PremiumResolution functionality.

**Valuespace:**
<S: 0..100>

**Example:**
```
xStatus SystemUnit Software OptionKeys PremiumResolution
!*s SystemUnit Software OptionKeys PremiumResolution: "true"
** end
```

**xStatus SystemUnit Hardware Module SerialNumber**
Serial number of the full system (box).

**Valuespace:**
<S: 0..100>

**Example:**
```
xStatus SystemUnit Hardware Module SerialNumber
!*s SystemUnit Hardware Module SerialNumber: "B1AD25A00002"
** end
```

**xStatus SystemUnit Hardware Module Identifier**
States hardware revision of full system.

**Valuespace:**
<S: 0..100>

**Example:**
```
xStatus SystemUnit Hardware Module Identifier
!*s SystemUnit Hardware Module Identifier: "1"
** end
```

**xStatus SystemUnit Hardware MainBoard SerialNumber**
Serial number of main board.

**Valuespace:**
<S: 0..100>

**Example:**
```
xStatus SystemUnit Hardware MainBoard SerialNumber
!*s SystemUnit Hardware MainBoard SerialNumber: "FW0528833"
** end
```
The System Unit status, cont...

**xStatus SystemUnit Hardware MainBoard Identifier**
States hardware revision of MainBoard.

**Valuespace:**
<S: 0,100>

**Example:**
```
xStatus SystemUnit Hardware MainBoard Identifier
  *s SystemUnit Hardware MainBoard Identifier: "101400-5 [06]"
** end
```

**xStatus SystemUnit Hardware VideoBoard SerialNumber**
Serial number of VideoBoard.

**Valuespace:**
<S: 0,100>

**Example:**
```
xStatus SystemUnit Hardware VideoBoard SerialNumber
  *s SystemUnit Hardware VideoBoard SerialNumber: "PH0534914"
** end
```

**xStatus SystemUnit Hardware VideoBoard Identifier**
States hardware revision of VideoBoard.

**Valuespace:**
<S: 0,100>

**Example:**
```
xStatus SystemUnit Hardware VideoBoard Identifier
  *s SystemUnit Hardware VideoBoard Identifier: "101410-4 [07]"
** end
```

**xStatus SystemUnit Hardware AudioBoard SerialNumber**
Serial number of AudioBoard.

**Valuespace:**
<S: 0,100>

**Example:**
```
xStatus SystemUnit Hardware AudioBoard SerialNumber
  *s SystemUnit Hardware AudioBoard SerialNumber: "TBD"
** end
```

**xStatus SystemUnit Hardware AudioBoard Identifier**
States hardware revision of AudioBoard.

**Valuespace:**
<S: 0,100>

**Example:**
```
xStatus SystemUnit Hardware AudioBoard Identifier
  *s SystemUnit Hardware AudioBoard Identifier: "101420-2 [No obj1.]
** end
```

**xStatus SystemUnit Hardware BootSoftware**
Boot software id.

**Valuespace:**
<S: 0,100>

**Example:**
```
xStatus SystemUnit Hardware BootSoftware
  *s SystemUnit Hardware BootSoftware: "U-Boot 2009.03-37"
** end
```

**xStatus SystemUnit State System**
System state.

**Valuespace:**
<InCall/Initialized/Initializing/Multisite/Sleeping>

**Example:**
```
xStatus SystemUnit State System
  *s SystemUnit State System: Initialized
** end
```

**xStatus SystemUnit State MaxNumberOfCalls**
Max number of calls supported.

**Valuespace:**
<0..5>

**Example:**
```
xStatus SystemUnit State MaxNumberOfCalls
  *s SystemUnit State MaxNumberOfCalls: 3
** end
```
The System Unit status, cont...

**xStatus SystemUnit State MaxNumberOfActiveCalls**
Max number of active calls supported.

**Valuespace:**
<0..5>

**Example:**
```c
xStatus SystemUnit State MaxNumberOfActiveCalls
*s SystemUnit State MaxNumberOfActiveCalls: 3
** end
```

**xStatus SystemUnit State NumberOfActiveCalls**
Number of active calls.

**Valuespace:**
<0..5>

**Example:**
```c
xStatus SystemUnit State NumberOfActiveCalls
*s SystemUnit State NumberOfActiveCalls: 0
** end
```

**xStatus SystemUnit State NumberOfSuspendedCalls**
Number of suspended calls.

**Valuespace:**
<0..5>

**Example:**
```c
xStatus SystemUnit State NumberOfSuspendedCalls
*s SystemUnit State NumberOfSuspendedCalls: 0
** end
```

**xStatus SystemUnit State Subsystem Application**
State of system application.

**Valuespace:**
<Initialized/Initializing>

**Example:**
```c
xStatus SystemUnit State Subsystem Application
*s SystemUnit State Subsystem Application: Initialized
** end
```

**xStatus SystemUnit ContactInfo**
This gives the address another system should dial to reach this system.

**Valuespace:**
<S: 0,.100>

**Example:**
```c
xStatus SystemUnit ContactInfo
*s SystemUnit ContactInfo: "support@company.com"
** end
```
**The Video Input status**

**xStatus Video Input**

Run this command to see an 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 2 Connected: False
*s Video Input DVI 2 Connected: Unknown
*s Video Input DVI 3 Connected: Unknown
*s Video Input Legacy 3 Connected: False
** end
```

**xStatus Video Input Source [1..3] Resolution Height**

Last detected resolution height for source.

**Valuespace:**

<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**

Last detected resolution width for source.

**Valuespace:**

<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**

Last detected resolution refresh rate for source.

**Valuespace:**

<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**

True if a connection on HDMI 1..2 is detected. Not all connections can be detected.

**Valuespace:**

<True/False>

**Example:**

```
xStatus Video Input HDMI 1 Connected
*s Video Input HDMI 1 Connected: True
** end
```

**xStatus Video Input HDMI [1..2] Connected**

True if a connection on HDMI 1..2 is detected. Not all connections can be detected.

**Valuespace:**

<True/False>

**Example:**

```
xStatus Video Input HDMI 1 Connected
*s Video Input HDMI 1 Connected: True
** end
```
The Video Input status, cont...

**xStatus Video Input DVI [2, 3] Connected**
True if a connection on DVI 3 or DVI 5 is detected. Not all connections can be detected.

Valuespace:

<True/False>

Example:

```
xStatus Video Input DVI 2 Connected
*s Video Input DVI 2 Connected: False
** end
```

**xStatus Video Input Legacy [5..5] Connected**
True if a connection on Legacy 5 is detected. Not all connections can be detected.

Valuespace:

<True/False>

Example:

```
xStatus Video Input Legacy 5 Connected
*s Video Input Legacy 5 Connected: False
** end
```

The Video Output status

**xStatus Video Output**
Run this command to see an 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 Legacy 3 Resolution Height: 480
*s Video Output Legacy 3 Resolution Width: 720
*s Video Output Legacy 3 Resolution RefreshRate: 60
** end
```

**xStatus Video Output HDMI [1..1] Resolution Height**
Last confirmed resolution height for output HDMI 1.

Valuespace:

<120..3000>

Example:

```
xStatus Video Output HDMI 1 Resolution Height
*s Video Output HDMI 1 Resolution Height: 720
** end
```

**xStatus Video Output HDMI [1..1] Resolution Width**
Last confirmed resolution width for output HDMI 1.

Valuespace:

<176..4000>

Example:

```
xStatus Video Output HDMI 1 Resolution Width
*s Video Output HDMI 1 Resolution Width: 1280
** end
```
The Video Output status, **cont**...

**xStatus Video Output HDMI [1..1] Resolution RefreshRate**
Last confirmed refresh rate for output HDMI 1.

*Valuespace:*

<1..300>

*Example:*

```c
xStatus Video Output HDMI 1 Resolution RefreshRate
*s Video Output HDMI 1 Resolution RefreshRate: 60
** end
```

**xStatus Video Output DVI [2..2] Resolution Height**
Last confirmed resolution height for output DVI 2.

*Valuespace:*

<120..3000>

*Example:*

```c
xStatus Video Output DVI 2 Resolution Height
*s Video Output DVI 2 Resolution Height: 720
** end
```

**xStatus Video Output DVI [2..2] Resolution Width**
Last confirmed resolution width for output DVI 2.

*Valuespace:*

<176..4000>

*Example:*

```c
xStatus Video Output DVI 2 Resolution Width
*s Video Output DVI 2 Resolution Width: 1280
** end
```

**xStatus Video Output DVI [2..2] Resolution RefreshRate**
Last confirmed refresh rate for output DVI 2.

*Valuespace:*

<1..300>

*Example:*

```c
xStatus Video Output DVI 2 Resolution RefreshRate
*s Video Output DVI 2 Resolution RefreshRate: 60
** end
```

**xStatus Video Output Legacy [5..5] Resolution Height**
Last confirmed resolution height for output Legacy 5.

*Valuespace:*

<120..3000>

*Example:*

```c
xStatus Video Output Legacy 5 Resolution Height
*s Video Output Legacy 5 Resolution Height: 480
** end
```

**xStatus Video Output Legacy [5..5] Resolution Width**
Last confirmed resolution width for output Legacy 5.

*Valuespace:*

<176..4000>

*Example:*

```c
xStatus Video Output Legacy 5 Resolution Width
*s Video Output Legacy 5 Resolution Width: 720
** end
```

**xStatus Video Output Legacy [5..5] Resolution RefreshRate**
Last confirmed refresh rate for output Legacy 5.

*Valuespace:*

<1..300>

*Example:*

```c
xStatus Video Output Legacy 5 Resolution RefreshRate
*s Video Output Legacy 5 Resolution RefreshRate: 60
** end
```
The Video Layout status

xStatus Video Layout
Run this command to see an overview of the video status.

Example:

```plaintext
xStatus Video Layout
*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 Site [1..n] Output [1..3] FamilyName
Specifies what layout family is used on the specific output.

Valuespace:

<s: 0..100>

Example:

```plaintext
xStatus Video Layout Site 1 Output 1 FamilyName
*s Video Layout Site 1 Output 1 FamilyName: “full”
```

** end

The Video Layout status, cont...

xStatus Video Layout Site [1..n] Output [1..3] FullFamilyName
Specifies family with selfview information is used on the specific output.

Valuespace:

<s: 0..100>

Example:

```plaintext
xStatus Video Layout Site 1 Output 1 FullFamilyName
*s Video Layout Site 1 Output 1 FullFamilyName: “full-sv-on”
```

** end

xStatus Video Layout Site [1..n] Output [1..3] FamilyNumber
Layout family number used on the specific output.

Valuespace:

<Integer value>

Example:

```plaintext
xStatus Video Layout Site 1 Output 1 FamilyNumber
*s Video Layout Site 1 Output 1 FamilyNumber: 1014
```

** end

xStatus Video Layout Site [1..n] Output [1..3] GraphicName
Name of the layout used right now at the specified output. Note that while the FamilyName is constant as long as configurations on the system doesn't change, the GraphicName varies depending on system state (how many participants for instance).

Valuespace:

<s: 0..100>

Example:

```plaintext
xStatus Video Layout Site 1 Output 1 GraphicName
*s Video Layout Site 1 Output 1 GraphicName: “full-pip”
```

** end
The Video Layout status, *cont...*

**xStatus Video Layout Site [1..n] Output [1..3] GraphicNumber**
A number identifying the layout used right now at the specified output.

Valuespace:

```
<Integer value>
```

Example:

```
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**
Descriptor of the chosen layout for the specified output

Valuespace:

```
<Integer value>
```

Example:

```
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**
DescriptorOutput of the chosen layout for the specified output

Valuespace:

```
<Integer value>
```

Example:

```
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**
The horizontal position of the upper left corner of a specific frame on a specific output.

Valuespace:

```
<0..10000>
```

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 PositionX
  "s Video Layout Site 1 Output 1 Frame 1 PositionX: 0
  ** end
```

**xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] PositionY**
The vertical position of the upper left corner of a specific frame on a specific output.

Valuespace:

```
<0..10000>
```

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 PositionY
  "s Video Layout Site 1 Output 1 Frame 1 PositionY: 0
  ** end
```

**xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] Width**
The width of a specific frame on a specific output.

Valuespace:

```
<0..10000>
```

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 Width
  "s Video Layout Site 1 Output 1 Frame 1 Width: 10000
  ** end
```

**xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] Height**
The height of a specific frame on a specific output.

Valuespace:

```
<0..10000>
```

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 Height
  "s Video Layout Site 1 Output 1 Frame 1 Height: 10000
  ** end
```

**xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] Layer**
The layer of a specific frame on a specific output.

Valuespace:

```
<1..3>
```

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 Layer
  "s Video Layout Site 1 Output 1 Frame 1 Layer: 1
  ** end
```
The Video Layout status, cont...

**xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] VideoSourceType**

Describes what type of content is displayed in a specific frame on a specific output.

*Valuespace:* 
\(<S:0,100>\)

*Example:*

```plaintext
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**

If the type of content that is displayed in a specific frame on a specific output need an identifier, this is shown here.

*Valuespace:* 
\(<1..3>\)

*Example:*

```plaintext
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**

If the type of content that is displayed in a specific frame on a specific output need an input number, this is shown here.

*Valuespace:* 
\(<1..3>\)

*Example:*

```plaintext
xStatus Video Layout Site 1 Output 1 Frame 1 InputNumber
*s Video Layout Site 1 Output 1 Frame 1 InputNumber: 0
** end
```

The Video Layout status, cont...

**xStatus Video Layout Site [1..n] Output [1..3] Frame [1..6] Filename**

If graphics is shown in a frame in a specific output, the filename where the graphics is located is specified here.

*Valuespace:* 
\(<S:0,200>\)

*Example:*

```plaintext
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 commands
The Experimental commands can be used ‘as is’ and will not be further documented.

NOTE! The Experimental commands WILL change.

**xStatus Experimental Audio Input Connectors Microphone [1..4] Activity**
The Experimental commands can be used ‘as is’. NOTE! The Experimental settings WILL change.
States whether there is a signal on the connector.

Valuespace:

\(<\text{True}/\text{False}>\)

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**
The Experimental commands can be used ‘as is’. NOTE! The Experimental settings WILL change.
States whether there is a signal on the connector.

Valuespace:

\(<\text{True}/\text{False}>\)

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**
The Experimental commands can be used ‘as is’. NOTE! The Experimental settings WILL change.
States whether there is a signal on the connector.

Valuespace:

\(<\text{True}/\text{False}>\)

Example:

```
xstatus Experimental Audio Input Connectors HDMI 3 Activity
> s Experimental Audio Input Connectors HDMI 3 Activity: False
** end
```
Chapter 6

Appendices
Dynamic audio API

With a set of xCommand Audio API commands, the audio system is fully configurable:

- A local input is defined as a mix of input connectors. A local input can be created and deleted. You can add and remove input connectors and update the local input settings.
- A local output is a mix of local input and remote input signals. All connectors attached to a local output receive the same signal.
- A remote input and remote output pair is created for each call.
- A remote output receives the signals from all local and remote inputs, except the remote input from the same site.
- The mixer matrix of each local and remote output can easily be managed by disconnecting and connecting local and remote inputs.

Example 1:
The dynamic audio API offers a great flexibility when configuring the system and it is simple to use. To have audio in your calls you only need one local input with a microphone attached to it, and one local output with an output connector attached to it.

When the call is established a remote input and remote output pair are created and all the connections to these from the local audio system are set up automatically.

<table>
<thead>
<tr>
<th>Local input 1</th>
<th>Local output 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MICROPHONE 1</td>
<td>Line 1</td>
</tr>
<tr>
<td>MICROPHONE 2</td>
<td>Line 2</td>
</tr>
</tbody>
</table>

The Local input 1 has been defined with two Microphones.

The Local output 2 has been defined with two Line outputs.

The audio from the Remote input 3 goes to the Local output 2.

The audio from the Local input 1 goes to the Remote output 4.
The equalizer

The system has 8 user defined equalizers, each made up of 6 second order IIR sections, see the illustration below. Each of these 8 equalizers can be applied to one or more of the audio input and output connectors on the codec.

The Audio Console application

We recommend using the Audio Console, with the build in equalizer GUI, to modify the equalizers. The Audio Console application is found at the TANDBERG Developer Zone web page. Go to: http://developer.tandberg.com/web/guest/tools/integrators/audio-console

The equalizer filter parameters

Each section (1 … 6) of each user defined equalizer (1 … 8) can be modified independently by setting the four parameters:

- Filter type (Peaking, low shelf, high shelf, low pass, high pass)
- Frequency
- Gain
- Q-value

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).

The filter types

The illustrations below shows the 5 different filter types and 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 bandwith in Hz.

The equalizer IIR filter

<table>
<thead>
<tr>
<th>Section 1</th>
<th>Section 2</th>
<th>Section 3</th>
<th>Section 4</th>
<th>Section 5</th>
<th>Section 6</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Diagram of Section 1" /></td>
<td><img src="image2" alt="Diagram of Section 2" /></td>
<td><img src="image3" alt="Diagram of Section 3" /></td>
<td><img src="image4" alt="Diagram of Section 4" /></td>
<td><img src="image5" alt="Diagram of Section 5" /></td>
<td><img src="image6" alt="Diagram of Section 6" /></td>
</tr>
</tbody>
</table>
Stereo

Stereo in point to point call

Receive stereo
If the other participant sends stereo the codec will receive stereo. To play stereo the local output still needs to be configured correctly (see local stereo circuit example).

Transmit stereo
In order to send stereo, the other participant must be able to receive stereo. Also, the local input with the stereo signal must be configured correctly (see local stereo circuit example). In addition, there must be a signal on one of the connectors connected to the local input.

Stereo in Multisite
The TANDBERG Codec C60 will only send and receive mono signals in multisite.

Example of local stereo configuration
When you want to record a stereo presentation the audio system can be configured as shown in the illustration.
In the example the local input has two connectors: Line 1 and Line 2. When these two connectors are mixed together as a stereo signal, Line 1 is the left channel and Line 2 is the right channel.
And the local output has two connectors: Line 1 and Line 2. The local output pans the stereo signal according to the channel configuration of the connectors. Line 1 receives the left channel and Line 2 receives the right channel.

Local stereo circuit example with API commands
The graphic gives a picture of how to configure the audio system for recording a stereo presentation. You can do the same from the API. The belonging xCommands and xConfigurations are listed below.

In this example there is a local input with input id 5 and a local output with output id 6. Both are set to stereo by the channels parameter.

- xCommand Audio LocalInput Update Inputid:5 Name: "Presentation" MixerMode:Fixed AGC:On Mute:Off Channels:2
- xCommand Audio LocalOutput Update Outputid:6 Name: "Recorder" Loudspeaker:No Channels:2
- xConfiguration Audio Input Line 1 Channel: Left
- xConfiguration Audio Input Line 2 Channel: Right
- xConfiguration Audio Output Line 1 Channel: Left
- xConfiguration Audio Output Line 2 Channel: Right
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.
Remote Control TRC5–
Key map

The TANDBERG Remote control TRC5 has the following button codes and IR signal parameters.

You will find a one page overview of the TANDBERG Remote Control TRC5 overleaf.

<table>
<thead>
<tr>
<th>Dec</th>
<th>Hex</th>
<th>Address</th>
<th>Button name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Number 1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>Number 2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0</td>
<td>Number 3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>Number 4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0</td>
<td>Number 5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>0</td>
<td>Number 6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>0</td>
<td>Number 7</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>0</td>
<td>Number 8</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>0</td>
<td>Number 9</td>
</tr>
<tr>
<td>10</td>
<td>0A</td>
<td>0</td>
<td>Number 0</td>
</tr>
<tr>
<td>11</td>
<td>0B</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0C</td>
<td>0</td>
<td>#</td>
</tr>
<tr>
<td>13</td>
<td>0D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>11</td>
<td>0</td>
<td>Presenter</td>
</tr>
<tr>
<td>18</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>16</td>
<td>0</td>
<td>Zoom out</td>
</tr>
<tr>
<td>23</td>
<td>17</td>
<td>0</td>
<td>Zoom in</td>
</tr>
<tr>
<td>24</td>
<td>18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>19</td>
<td>0</td>
<td>Volume down</td>
</tr>
<tr>
<td>26</td>
<td>1A</td>
<td>0</td>
<td>Volume up</td>
</tr>
<tr>
<td>27</td>
<td>1B</td>
<td>0</td>
<td>Microphone off</td>
</tr>
<tr>
<td>28</td>
<td>1C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>1D</td>
<td>0</td>
<td>Arrow up</td>
</tr>
<tr>
<td>30</td>
<td>1E</td>
<td>0</td>
<td>Arrow down</td>
</tr>
<tr>
<td>31</td>
<td>1F</td>
<td>0</td>
<td>Arrow left</td>
</tr>
<tr>
<td>32</td>
<td>20</td>
<td>0</td>
<td>Arrow right</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dec</th>
<th>Hex</th>
<th>Address</th>
<th>Button name</th>
</tr>
</thead>
<tbody>
<tr>
<td>33</td>
<td>21</td>
<td>0</td>
<td>Ok</td>
</tr>
<tr>
<td>34</td>
<td>22</td>
<td>0</td>
<td>Call</td>
</tr>
<tr>
<td>35</td>
<td>23</td>
<td>0</td>
<td>End call</td>
</tr>
<tr>
<td>36</td>
<td>24</td>
<td>0</td>
<td>Phone book</td>
</tr>
<tr>
<td>37</td>
<td>25</td>
<td>0</td>
<td>Layout</td>
</tr>
<tr>
<td>38</td>
<td>26</td>
<td>0</td>
<td>Clear</td>
</tr>
<tr>
<td>39</td>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>2A</td>
<td>0</td>
<td>Soft key 1</td>
</tr>
<tr>
<td>43</td>
<td>2B</td>
<td>0</td>
<td>Soft key 2</td>
</tr>
<tr>
<td>44</td>
<td>2C</td>
<td>0</td>
<td>Soft key 3</td>
</tr>
<tr>
<td>45</td>
<td>2D</td>
<td>0</td>
<td>Soft key 4</td>
</tr>
<tr>
<td>46</td>
<td>2E</td>
<td>0</td>
<td>Soft key 5</td>
</tr>
<tr>
<td>47</td>
<td>2F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>33</td>
<td>0</td>
<td>Home</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td></td>
</tr>
<tr>
<td>Reference frequency</td>
<td>-</td>
</tr>
<tr>
<td>Address</td>
<td>-</td>
</tr>
<tr>
<td>IR wavelength</td>
<td>-</td>
</tr>
<tr>
<td>IR carrier frequency</td>
<td>-</td>
</tr>
</tbody>
</table>
Remote Control TRC5—One page overview

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 Up/Down arrow key to adjust the codec volume.
- **Mute:** Press the – on the Up/Down arrow 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 show the menu on screen.

- **Call:** Using the key:
  - **INITIATE A CALL:** Select a name from the Phone book or enter the name, number or URI and press the Call key to initiate the call.
  - **SHORTCUT TO RECENT CALLS:** Use the Call key as a shortcut to Recent Calls when the Call menu is not visible.

**Clear:** Press the C key to remove characters in a text field.

**Waking up the system**
Grab the remote control and make sure your hand touches the rubber line sensors going along both sides of the remote control.

or: Touch any key on the remote control.

**Presentation:** Press the key to show/hide a presentation.

**Zoom:** Press the + or – on the Up/Down arrow key to zoom the camera in and out.

**Arrows:**
- **Up/Down:** Use the Up/Down 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.