

Software version IL1.0
JUNE 2012



Application Programmer Interface (API) Reference Guide

Cisco TelePresence ISDN Link

What's in this guide?

The top menu bar and the entries in the Table of Contents are all hyperlinks, just click on them to go to the topic.

We recommend you visit our web site regularly for updated versions of the user documentation. Go to: ► <http://www.cisco.com/go/isdnlink-docs>

Table of Contents

Introduction

- About this guide 4
- User documentation 4
- Software 4
- Cisco contacts..... 4

About the API

- About the API 6
- The API-Engine 6
- Structuring of Information..... 6
- Addressing Using XPath or SimplePath 7
- Feedback..... 7

- Connecting to the ISDN Link 8
- Accessing XACLI..... 8
- COM Port settings..... 8
 - SSH login 8
 - Serial port login..... 8
- Hardware & Cabling (RS-232) 8
- Troubleshooting (RS-232) 8

- Value types and formats 9

- User commands 10

- Main type of commands..... 11
 - Configuration type commands 11
 - Command type commands 11
 - Status type commands 11
 - Special commands 11
 - Feedback type command..... 11
 - Preferences type command 11

- About xConfiguration 12

- About xCommand 13

- About xStatus commands 14

- About xFeedback 15

- About xPreferences 16

- The SystemTools commands 17

Description of the xConfiguration commands

- Description of the xConfiguration commands 20

Description of the xCommand commands

- Description of the xCommands commands..... 38

Description of the xStatus commands

- Description of the xStatus commands 42

Chapter 1

Introduction

About this guide

This guide will introduce you to the Application Programmer Interface (API) and serve as a reference guide for the command line commands for Cisco TelePresence ISDN Link.

User documentation

The user documentation for the Cisco TelePresence ISDN Link, running the **IL software**, have several guides.

- Installation guide
- Software release notes
- Administrator guide
- API reference guide
- Regulatory compliance and safety information
- License information

Download the user documentation

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

Software

You can download the software for your product from the Cisco web site.

Go to: ► <http://www.cisco.com/cisco/software/navigator.html>

Cisco contacts

On our web site you will find an overview of the worldwide Cisco contacts.

Go to: ► <http://www.cisco.com/web/siteassets/contacts>

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

Chapter 2

About the API

About the API

The API-engine can be accessed by an easy-to-use Command Line Interface called XACLI using RS-232 or SSH.

The information is stored in a hierarchic tree structure which is accessible from different interfaces.

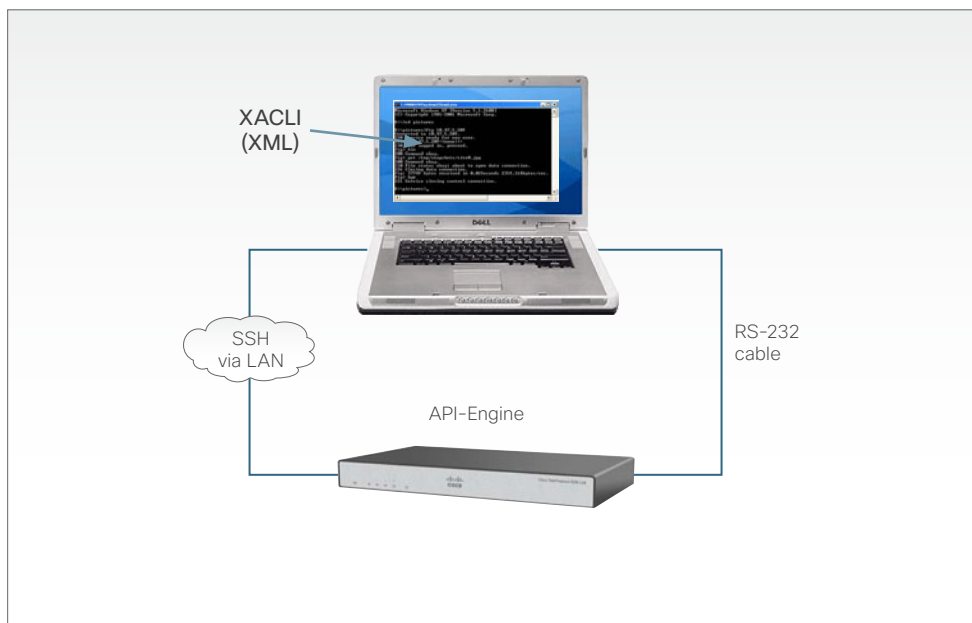
- When accessing the API-engine using XACLI (RS-232 or SSH), the information is formatted in a proprietary Command Line style or in XML formatting.

The API-Engine

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

The main features can be summarized to:

- Structuring of information
- Addressing using XPath (XML Path Language) or SimplePath
- 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:

- Information flowing from target. This we call **READ** information (**R**). The (**R**) should not be confused with the (**r**) used to indicate required parameters for xCommand.
- Information flowing to target. This we call **WRITE** information (**W**).

Main types of information

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

(R) READ information: xStatus.

Typical examples include: read the network status, ISDN 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: xCommand.

Typical examples include: set date and time, get date and time, software upgrade, 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: xConfiguration.

Typical examples include: set or read the value of the H320 network type, ISDN PRI switch type, call rate, etc. All configuration information is structured in a hierarchy making up a database of system settings.

Addressing Using XPath or SimplePath

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

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

Using XPath

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

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

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

Example:

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

Using SimplePath

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

Both levels and item numbers are separated with white spaces:

- Network 1 DNS Server 1 Address

Example:

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

Feedback

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

The user can specify what parts of the status and configuration hierarchies it wants to monitor by using XPath. The user 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 XACLI (RS-232/SSH).

The system uses SimplePath when presenting configurations.

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

Connecting to the ISDN Link

Accessing XACLI

XACLI can be accessed through 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 unit. The connector is marked with the text: **COM**.

COM Port settings

Use: 115200 bps, 8 data bits, 1 stop bit, no parity.

SSH login

- User name is: admin
- The default password is blank.

Serial port login

- User name is: admin
- The default password is blank.

Hardware & Cabling (RS-232)

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

Cable. A straight-through cable should be used between the RS-232 port and the DTE. The lower table shows the recommended cable-wiring scheme when connecting the ISDN Link 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 unit, the following should be checked:

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

COM port (RS-232)		
Pin	Signal name	Direction
1	Carrier detect, CD	From DCE
2	Receive data, RXD	From DCE
3	Transmit data, TXD	To DCE
4	Data terminal ready, DTR	From DCE
5	Signal GND	
6	Data set ready, DSR	From DCE
7	Ready to send, RTS	To DCE
8	Clear to send, CTS	From DCE
9	Ring indicator, RI	From DCE

Cable wiring (RS-232) DCE <-> PC		
DCE 9 pin	Direction	PC DTE, 9 pin
1 CD	→	1 CD
2 RD	→	2 RD
3 TD	←	3 TD
4 DTR	←	4 DTR
5 GND	↔	5 GND
6 DSR	→	6 DSR
7 RTS	←	7 RTS
8 CTS	→	8 CTS
9 RI	→	9 RI

Value types and formats

The system supports the following value types:

- Integer values
- Literal values
- String values

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

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

User commands

By typing [?](#) or [help](#) after connecting to the unit, the system will list all supported root commands.

Bye

The bye command will close the command line interface.

Echo <on/off>

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

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

The other commands

The other user commands are described in the following pages.

```
?  
      - User Commands -  
  
help          xcommand      xconfiguration  xfeedback      xgetxml  
xpreferences  xstatus       bye             echo           log  
systemtools  
OK
```

Main type of commands

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

The main types are:

- Configuration type commands
- Status type commands
- Command type commands

Configuration type commands

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

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

Example: Configure ISDN settings, external network settings, 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 commands are supplied by the user.

Actions performed by a Command type command are not persistent over a reboot.

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

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

Supported Command type commands:

- ▶ xCommand

Status type commands

Status type commands returns information about the system and system processes. xStatus 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

Special commands

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

Feedback type command

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

More on this can be found in xfeedback.

- ▶ xFeedback

Preferences type command

The Preference type command is used to set various preferences for the RS-232/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 system very simple.

More on this can be found in xpreferences.

- ▶ xPreferences

About xConfiguration

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

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

- `xConfiguration ?`
- `xConfiguration help`

To get an overview of the supported xConfiguration commands, with the corresponding value space, type `??` 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

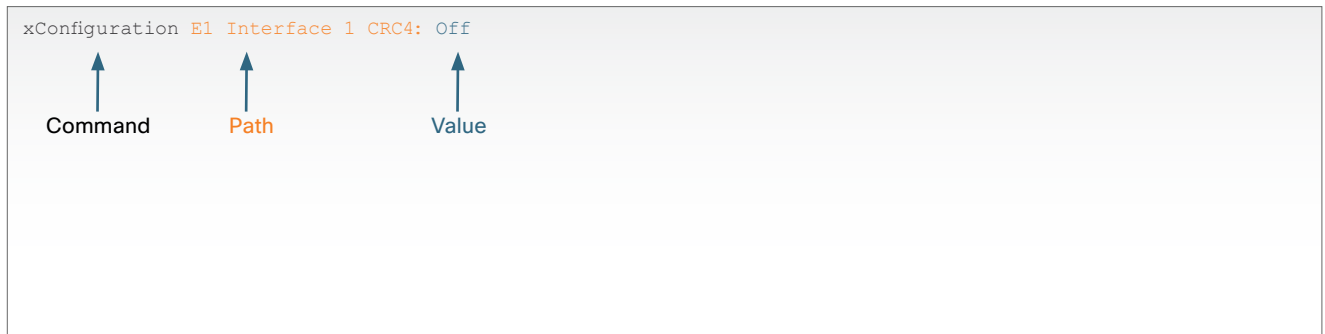
```
xconfiguration ?
- User Configurations -

E1                H320                Security           Time
Experimental      ISDN                SIP
ExternalNetwork   Network             SystemUnit
Gateway           NetworkServices    T1

OK
```

```
xConfiguration ??

*? xConfiguration E1 Interface [1..1] CRC4: <Off/On>
*? xConfiguration ExternalNetwork Interface [1..1] DtrPulse: <Off/On>
*? xConfiguration ExternalNetwork Interface [1..1] Clocking: <Dual/Single>
*? xConfiguration ExternalNetwork Interface [1..1] CallControl: <Manual/RS366/RS366AdtranIMUX/RS366CustomIMUX/LeasedLine>
*? xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw64 Prefix: <S: 0, 12>
*? xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw64 Suffix: <S: 0, 12>
*? xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw64R Prefix: <S: 0, 12>
*? xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw64R Suffix: <S: 0, 12>
.
.
.
OK
```



About xCommand

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

To get an overview of the supported xCommand type commands, type [?](#) or [help](#) after the command:

- xCommand ?
- xCommand help

To get an overview of the supported xCommand type commands with the corresponding value space, type [??](#) after the command:

- xCommand ??

Required parameters are identified by an **(r)** behind the parameter name.

When issuing a xCommand type command, the command consists of several parts:

1. The type of command: xCommand
2. The path: An address expression
3. The parameter: The parameter, terminated by a colon, followed the corresponding value.
4. If more parameters: The parameter, terminated by a colon, followed by the corresponding value.

```
xcommand ?
- User Commands -

Boot          Security      SystemUnit

OK
```

```
xcommand ??
xCommand Boot

xCommand Security FipsMode Activate
  Confirm(r): <Yes>

xCommand SystemUnit AdminPassword Set
  Password(r): <S: 0, 64>
xCommand SystemUnit DateTime Set
  Year: <2008..2037>
  Month: <1..12>
  Day: <1..31>
  Hour: <0..23>
  Minute: <0..59>
  Second: <0..59>
xCommand SystemUnit DateTime Get
xCommand SystemUnit FactoryReset
  Confirm(r): <Yes>
xCommand SystemUnit SoftwareUpgrade
  URL(r): <S: 0, 255>
  UserName: <S: 0, 255>
  Password: <S: 0, 255>

OK
```



About xStatus commands

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

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

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

- `xStatus ?`
- `xStatus help`

When issuing a xStatus command, the command consists of several parts. You can query all information or just some of it.

1. The type of command: xCommand
2. The path: An address expression with one or more items of the path included.

```
xStatus ?
- Status -

Call[...]          H320          MediaChannels      SIP
ExternalNetwork    ISDN          Network            SystemUnit

OK
```

```
Example with only one item of the path.
xStatus Network
*s Network 1 Ethernet MacAddress: "00:50:60:06:C5:52"
*s Network 1 Ethernet Link Network: "1000full"
*s Network 1 Ethernet Link Endpoint: "No LAN"
*s Network 1 IPv4 Address: "192.168.1.0"
*s Network 1 IPv4 SubnetMask: "255.255.255.0"
*s Network 1 IPv4 Gateway: "192.168.1.1"
*s Network 1 IPv4 DNS Domain Name: "company.com"
*s Network 1 IPv4 DNS Server 1 Address: "192.168.10.10"
*s Network 1 IPv4 DNS Server 2 Address: "192.168.11.11"
*s Network 1 IPv4 DNS Server 3 Address: ""
*s Network 1 IPv4 DNS Server 4 Address: ""
*s Network 1 IPv4 DNS Server 5 Address: ""
*s Network 1 IPv6 Address: ""
*s Network 1 IPv6 Gateway: ""
*s Network 1 MTU: 1500
** end

OK
```

```
Example with more than one item of the path.
xStatus Network 1 Ethernet Link
*s Network 1 Ethernet Link Network: "1000full"
*s Network 1 Ethernet Link Endpoint: "No LAN"
```

About xFeedback

The xFeedback is a feature that lets you subscribe to notifications of changes from the system.

- This can be xConfiguration, xStatus changes and changes in the state of the unit.

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

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

CAUTION: We discourage registering all status changes as this may give more feedback information than the system is able to handle.

```
xFeedback ?
```

```
xFeedback help:
```

<code>xFeedback Register XPathExpression[1..255]</code>	- Registers feedback on expression XPathExpression
<code>xFeedback Deregister XPathExpression[1..255]</code>	- Deregisters feedback if registered on XPathExpression
<code>xFeedback deregisterall</code>	- Deregister all expressions
<code>xFeedback List</code>	- Generate list of currently registered XPathExpressions
<code>xFeedback Help</code>	- Display this help text

Example with xFeedback

```
xFeedback register Status/ISDN
xFeedback register Configuration/ISDN

xFeedback list
xFeedback deregister Configuration/ISDN
xFeedback list
```

About xPreferences

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

```
xPreferences ?

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


The SystemTools commands

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

Supported commands

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

Example:

```
systemtools ?
bondingtrace
dumph221
isdntrace
license
network
passwd
rootsettings
securitysettings
securitystatus
showlog
OK
```

Parameters

Required parameters in angle brackets: <text>

Optional parameters in square brackets: [text]

Detailed information

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

Example:

```
systemtools license ?
usage: license <list | show <name>>
OK
```

systemtools bondingtrace on|off

Control logging of misc. bonding related events. Logging is typically sent to the application log file, but can be enabled for a console by enabling 'log' output, see separate command for this. Please notice that the bonding log may interfere with real time characteristics, and increases the chance of loosing H.320 side traffic data.

on/off: Enable/disable decoding trace output.

systemtools dumph221 [inst] toggle

Dump H.221 internal information used for debugging purposes.

inst: H.221 call instance (0-3), defaults to 0.

toggle: Toggle continuous dump on/off.

systemtools dumph221 reset

Reset H.221 dump buffer.

systemtools isdntrace

List current trace settings for logging of the ISDN D-channel signalling.

systemtools isdntrace decode on|off

Logging of the ISDN D-channel signalling.

on/off: Enable/disable decoding trace output.

systemtools sdntrace on|off [bri|pri [if]]

Logging of the ISDN D-channel signalling.

on/off: Enable/disable tracing.

bri/pri: Select the type of ISDN.

if: Define which interface to enable/disable trace for. Interfaces are numbered from 1. If not specified, all interfaces selected.

systemtools isdntrace output on|off

Logging of the ISDN D-channel signalling.

on/off: Enable/disable trace output for console.

systemtools license list

Lists all the licenses for the ISDN Link.

systemtools license show <name>

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

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

systemtools network ping <hostname>

Network debug command.

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

systemtools network traceroute <hostname>

Network debug command.

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

systemtools network netstat

Network debug command.

systemtools passwd

Change the password for the logged in user. When prompted enter the current password and the new password, and confirm the new password.

systemtools rootsettings get

Obtain the current setting for the systemtools rootsetting.

systemtools rootsettings on [password]

Command to control the root user availability and set the root password. Enable access to the system for the root user on all ports.

password: The root user password. Use "" as password to clear root password.

systemtools rootsettings serial [password]

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

password: The root user password.

systemtools rootsettings off

Command to control the root user availability.

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

systemtools rootsettings never

Command to control the root user availability.

NOTE! The root user is permanently turned off!

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

systemtools securitysettings jitc

Set up security requirements so they meet JITC.

Set or view password and PIN polices enforced on the unit.

systemtools securitysettings isjitc

Check if the current settings are JTIC compliant.

systemtools securitysettings default

Revert to default security settings.

systemtools securitysettings ask

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

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

Max failed login attempts [0]?

- Number of failed log ins until a user is set inactive.

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

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

Max simultaneous sessions total [0]?

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

Max simultaneous sessions per user [0]?

- Maximum number of simultaneous sessions per user.

Number of passwords to remember [0]?

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

Number of PINs to remember [0]?

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

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

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

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

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

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

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

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

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

Maximum time between log ins (days) [0]?

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

Max consecutive equal digits in PINs [0]?

- Maximum consecutive equal digits in PINs.

Minimum number of digits in PINs [0]?

- Minimum number of digits in PINs.

Maximum number of digits in PINs [0]?

- Maximum number of digits in PINs.

Max consecutive identical characters in passwords [0]?

- Maximum consecutive identical characters in passwords.

Minimum number of characters in passwords [0]?

- Minimum number of characters in passwords.

Maximum number of characters in passwords [0]?

- Maximum number of characters in passwords.

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

- Minimum number of lower-case letters in passwords.

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

- Minimum number of upper-case letters in passwords.

Minimum number of numerical characters in passwords [0]?

- Minimum number of numerical characters in passwords.

Minimum number of special characters in passwords [0]?

- Minimum number of special characters in passwords.

Minimum number of character groups in passwords [0]?

- Minimum number of character groups in passwords.

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

- Minimum number of character changed from previous password.

systemtools securitystatus

Shows the security status for the ISDN Link.

systemtools showlog [n]

List available logs.

n: select archived log number 'n', where n=1 is the most recent.

systemtools showlog logname|all [n]

Show specific log or all logs.

logname/all: Specify the name of the log or select all.

n: select archived log number 'n', where n=1 is the most recent.

Chapter 3

Description of the xConfiguration commands

Description of the xConfiguration commands

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

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

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

The E1 configuration

xConfiguration E1 Interface [1..1] CRC4

E1 CRC-4 (Cyclic Redundancy Check 4-bit) is used for most E1-PRI configurations.

Value space: <On/Off>

On: The default value is On.

Off: Some manufactures do not support this feature. By setting E1 CRC4 to Off this functionality will be disabled.

Example: `xConfiguration E1 Interface 1 CRC4: On`

The ExternalNetwork configuration

xConfiguration ExternalNetwork Interface [1..1] DtrPulse

Configures the DTR signal on the External Network port (Net).

Value space: <On/Off>

On: The DTR signal will give a low pulse lasting for 5 seconds.

Off: The DTR pulse will stay low.

Example: `xConfiguration ExternalNetwork Interface 1 DtrPulse: Off`

xConfiguration ExternalNetwork Interface [1..1] Clocking

Configure the clocking signal which is compatible with the external equipment.

Value space: <Dual/Single>

Dual: (RS449/V35 Compatible) Use this setting when the external equipment provides two clock signals, one for transmit and one for receive. The difference between RS449 and V35 lies solely in the cable.

Single: (X21 Compatible) Use this setting when the external equipment provides a common clock signal for both transmit and receive.

Example: `xConfiguration ExternalNetwork Interface 1 Clocking: Dual`

xConfiguration ExternalNetwork Interface [1..1] CallControl

Configure the dialling sheme which is compatible with the external equipment.

Value space: <Manual/RS366/RS366AdtranIMUX/RS366CustomIMUX/LeasedLine>

Manual: Select Manual used when no handshake signals are available and the external equipment requires a constantly connected line.

RS366: This is the only dialling protocol supported and would normally be used together with Dual network clocking when the external equipment uses RS366 ports.

RS366AdtranIMUX: This setting offers extra usability when dialling RS366 via an ADTRAN IMUX. This dialling scheme will map the call type and bandwidth selection to ADTRAN IMUX specific suffixes to the dialled number. Should only be used when connected to an ADTRAN IMUX. The Adtran IMUX uses the following suffixes <Number>#C#R, where #C = Call Type #2 = audio #3 = 56kbps #4 = 64kbps #R = Channel Rate #0 = 2xh221 (2x56\64kbps) #1 to 8 = the Call Rate.

RS366CustomIMUX: Uses a custom prefix/suffix table which describes the available bandwidths. The prefixes/suffixes are set from the Web Interface or Command Line interface. The user (administrator) shall be able to specify a IMUX prefix/suffix table for the following bandwidths (kbps): 64, 64 Restrict, 128, 128 Restrict, 192, 192 Restrict, 256, 256 Restrict, 320, 320 Restrict, 384, 384 Restrict, 512, 512 Restrict, 768, 768 Restrict, 1152, 1152 Restrict, 1472, 1472 Restrict, 1920, 1920 Restrict.

LeasedLine: Leased Line is a non-dialling protocol and should be used when two codecs are connected in a point-to-point connection. Use Leased Line when the handshaking signals DTR and CD are available. DTR and CD correspond to the X.21 network's C and I signals.

Example: `xConfiguration ExternalNetwork Interface 1 CallControl: RS366`

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw64 Prefix

Defines the prefix to be used with the RS366 Custom IMUX at this bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw64 Prefix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw64 Suffix

Defines the suffix to be used with the RS366 Custom IMUX at this bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw64 Suffix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw64R Prefix

Defines the prefix to be used with the RS366 Custom IMUX at this restricted bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw64R Prefix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw64R Suffix

Defines the suffix to be used with the RS366 Custom IMUX at this restricted bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw64R Suffix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw128 Prefix

Defines the prefix to be used with the RS366 Custom IMUX at this bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw128 Prefix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw128 Suffix

Defines the suffix to be used with the RS366 Custom IMUX at this bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw128 Suffix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw128R Prefix

Defines the prefix to be used with the RS366 Custom IMUX at this restricted bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw128R Prefix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw128R Suffix

Defines the suffix to be used with the RS366 Custom IMUX at this restricted bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw128R Suffix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw192 Prefix

Defines the prefix to be used with the RS366 Custom IMUX at this bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw192 Prefix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw192 Suffix

Defines the suffix to be used with the RS366 Custom IMUX at this bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw192 Suffix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw192R Prefix

Defines the prefix to be used with the RS366 Custom IMUX at this restricted bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw192R Prefix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw192R Suffix

Defines the suffix to be used with the RS366 Custom IMUX at this restricted bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw192R Suffix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw256 Prefix

Defines the prefix to be used with the RS366 Custom IMUX at this bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw256 Prefix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw256 Suffix

Defines the suffix to be used with the RS366 Custom IMUX at this bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw256 Suffix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw256R Prefix

Defines the prefix to be used with the RS366 Custom IMUX at this restricted bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw256R Prefix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw256R Suffix

Defines the suffix to be used with the RS366 Custom IMUX at this restricted bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw256R Suffix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw320 Prefix

Defines the prefix to be used with the RS366 Custom IMUX at this bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw320 Prefix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw320 Suffix

Defines the suffix to be used with the RS366 Custom IMUX at this bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw320 Suffix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw320R Prefix

Defines the prefix to be used with the RS366 Custom IMUX at this restricted bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw320R Prefix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw320R Suffix

Defines the suffix to be used with the RS366 Custom IMUX at this restricted bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw320R Suffix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw384 Prefix

Defines the prefix to be used with the RS366 Custom IMUX at this bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw384 Prefix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw384 Suffix

Defines the suffix to be used with the RS366 Custom IMUX at this bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw384 Suffix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw384R Prefix

Defines the prefix to be used with the RS366 Custom IMUX at this restricted bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw384R Prefix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw384R Suffix

Defines the suffix to be used with the RS366 Custom IMUX at this restricted bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw384R Suffix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw512 Prefix

Defines the prefix to be used with the RS366 Custom IMUX at this bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw512 Prefix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw512 Suffix

Defines the suffix to be used with the RS366 Custom IMUX at this bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw512 Suffix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw512R Prefix

Defines the prefix to be used with the RS366 Custom IMUX at this restricted bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw512R Prefix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw512R Suffix

Defines the suffix to be used with the RS366 Custom IMUX at this restricted bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw512R Suffix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw768 Prefix

Defines the prefix to be used with the RS366 Custom IMUX at this bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw768 Prefix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw768 Suffix

Defines the suffix to be used with the RS366 Custom IMUX at this bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw768 Suffix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw768R Prefix

Defines the prefix to be used with the RS366 Custom IMUX at this restricted bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw768R Prefix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw768R Suffix

Defines the suffix to be used with the RS366 Custom IMUX at this restricted bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw768R Suffix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw1152 Prefix

Defines the prefix to be used with the RS366 Custom IMUX at this bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw1152 Prefix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw1152 Suffix

Defines the suffix to be used with the RS366 Custom IMUX at this bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw1152 Suffix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw1152R Prefix

Defines the prefix to be used with the RS366 Custom IMUX at this restricted bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw1152R Prefix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw1152R Suffix

Defines the suffix to be used with the RS366 Custom IMUX at this restricted bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: xConfiguration ExternalNetwork Interface 1 CustomIMUX bw1152R Suffix: ""

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw1472 Prefix

Defines the prefix to be used with the RS366 Custom IMUX at this bandwidth..

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: `xConfiguration ExternalNetwork Interface 1 CustomIMUX bw1472 Prefix: ""`

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw1472 Suffix

Defines the suffix to be used with the RS366 Custom IMUX at this bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: `xConfiguration ExternalNetwork Interface 1 CustomIMUX bw1472 Suffix: ""`

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw1472R Prefix

Defines the prefix to be used with the RS366 Custom IMUX at this restricted bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: `xConfiguration ExternalNetwork Interface 1 CustomIMUX bw1472R Prefix: ""`

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw1472R Suffix

Defines the suffix to be used with the RS366 Custom IMUX at this restricted bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: `xConfiguration ExternalNetwork Interface 1 CustomIMUX bw1472R Suffix: ""`

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw1920 Prefix

Defines the prefix to be used with the RS366 Custom IMUX at this bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: `xConfiguration ExternalNetwork Interface 1 CustomIMUX bw1920 Prefix: ""`

xConfiguration ExternalNetwork Interface [1..1] CustomIMUX bw1920 Suffix

Defines the suffix to be used with the RS366 Custom IMUX at this bandwidth.

Value space: <S: 0, 12>

Format: String with a maximum of 12 characters.

Example: `xConfiguration ExternalNetwork Interface 1 CustomIMUX bw1920 Suffix: ""`

The Gateway configuration

xConfiguration Gateway SIP PeerUri

Set the address be used to reach the peer codec for incoming H.320 side calls when SIP (VCS) registered mode is used. It should hold the same setting as the peer codec uses to register itself on the VCS server. If SIP registered mode is not used, this setting is not used. (VCS = Video Communication Server)

Value space: <S: 0, 256>

Format: Compact string with a maximum of 256 characters.

Example: `xConfiguration Gateway SIP PeerUri: "sip-codec@domain.com"`

xConfiguration Gateway SIP PeerHost

Set the IP address of the peer SIP codec using the ISDN link. Only one SIP codec can use the ISDN link, and it must be on the same subnet as the ISDN link. This IP address must correspond to the IP address type (IPv4 or IPv6) that the codec is configured to use for SIP calls.

The address is used for:

- 1) Verification of the SIP remote media transport addresses (if there is a mismatch, the call will be rejected/disconnected).
- 2) System status monitoring, indicating alarm etc if specified codec is not reachable.
- 3) If not SIP (VCS) registered, this address is used to call the SIP codec for incoming H.320 side calls. (VCS = Video Communication Server)

Value space: <S: 0, 64>

Format: Compact string with a maximum of 64 characters.

Example: `xConfiguration Gateway SIP PeerHost: "10.0.0.100"`

The H320 configuration

xConfiguration H320 NetType

Defines the network type to use for H.320 calls.

Value space: <BRI/PRI/External/G703>

BRI: Select BRI for the BRI network.

PRI: Select PRI for the PRI network.

External: Select External for the external network (Net/V.35).

G703: For test purpose only.

Example: `xConfiguration H320 NetType: BRI`

xConfiguration H320 Bonding Timer

Relaxed bonding timer is used when B channels requires additional time before they can become transparent.

Value space: <Normal/Relaxed>

Normal: The default value is Normal.

Relaxed: Relaxed bonding timing should be used with applications where the B channels use some additional time before they become transparent, like external encryption devices etc.

Example: `xConfiguration H320 Bonding Timer: Normal`

xConfiguration H320 Bonding Rebonding

Rebonding is used to re-establish H.320 calls if corrupted data is received for a longer period (10 - 15 sec).

Value space: <On/Off>

On: The default value is On.

Off: Some manufactures do not support this feature. By setting Rebonding to Off this functionality will be disabled.

Example: `xConfiguration H320 Bonding Rebonding: On`

xConfiguration H320 DefaultCall Restrict

A restricted call uses 56 kbps channels rather than the default unrestricted 64 kbps channels. Some older networks (primarily in the USA) do not support 64 kbps channels and require the use of restricted 56 kbps calls. By default, the system will dial an unrestricted call and downspeed to 56 kbps if necessary.

Value space: <On/Off>

On: Set to On to force restricted (56 kbps) calls.

Off: Set to Off to disable restricted calls.

Example: `xConfiguration H320 DefaultCall Restrict: Off`

The ISDN configuration

xConfiguration ISDN SendComplete

Select if the ISDN message should contain sending complete information.

Value space: <On/Off>

On: If set to On the system will send the ISDN message information element Sending Complete.

Off: If set to Off The system will not send Sending Complete.

Example: `xConfiguration ISDN SendComplete: Off`

xConfiguration ISDN SendNumber

Select if the system should send its own number to the far end.

Value space: <On/Off>

On: When set to On, the system will send its own numbers to the far end.

Off: When set to Off, the system will not send its own numbers to the far end. Note that the network may still send your numbers to the far end.

Example: `xConfiguration ISDN SendNumber: On`

xConfiguration ISDN ParallelDial

Select if the channels in bonding calls should be connected in parallel or one by one.

Value space: <On/Off>

On: When set to On, channels will be dialled and connected in parallel when setting up a bonding call.

Off: When set to Off, channels will be dialled one by one, which may increase the dialling time.

Example: `xConfiguration ISDN ParallelDial: On`

xConfiguration ISDN HCL

Select if HCL (Higher Level Capability) information should be included in the setup message (video calls only).

Value space: <On/Off>

On: When set to On, the system will signal, type of call i.e. video call, telephony or data. HLC must be enabled to work with some PRI networks in Italy.

Off: When set to Off, no HCL information will be sent.

Example: `xConfiguration ISDN HCL: Off`

xConfiguration ISDN SpeechTimers

Select if speech timers should be active for ISDN Layer 3, telephony calls.

Value space: <On/Off>

On: Set to On to enable speech timers (T310, T304, T301).

Off: When set to Off, there will be no T310, T304, T301.

Example: `xConfiguration ISDN SpeechTimers: On`

xConfiguration ISDN MSN

The use of MSN (Multiple Subscriber Number) enables the system to attach different ISDN terminals, with different numbers, to the same physical ISDN telephone line. This service can be ordered from your telephone company.

Value space: <On/Off>

On: Set to On to enable MSN on the video system when you have subscribed to a MSN service from your telephone company.

Off: When set to Off, only calls to numbers specified for the specific ISDN interfaces will be answered.

Example: `xConfiguration ISDN MSN: Off`

xConfiguration ISDN SubAddress

Using a sub address enables you to connect up to eight ISDN terminals to the same ISDN telephone number and line. The terminals are addressed by using different sub addresses. To call a terminal with a sub address, separate the ISDN telephone number and the sub address with a * (asterisk). Note that this service has limited access on some ISDN networks.

Value space: <s: 0, 20>

Format: String with a maximum of 20 characters.

Example: `xConfiguration ISDN SubAddress: ""`

xConfiguration ISDN SubAddressSend

Using a sub address enables you to connect up to eight ISDN terminals to the same ISDN telephone number and line. The terminals are addressed by using different sub addresses. To call a terminal with a sub address, separate the ISDN telephone number and the sub address with a * (asterisk). Note that this service has limited access on some ISDN networks.

Value space: <On/Off>

On: When set to on, the system will use the values set by "ISDN SubAddress".

Off: Set to Off to disable use of sub addresses.

Example: `xConfiguration ISDN SubAddressSend: On`

xConfiguration ISDN CliNumSpec

Select if calling party numbers should be enabled. Explicitly specify Calling Party Number fields Type/Plan to be used in outgoing calls in the ISDN setup message.

Value space: <On/Off>

On: When set to on, the system will use the values set by "ISDN CliNumPlan" and "ISDN CliNumType".

Off: When set to Off, default values are used.

Example: `xConfiguration ISDN CliNumSpec: Off`

xConfiguration ISDN CliNumType

Value to be used for calling party number field Type. If the value is set to 2, the number type in the setup message will be set to National. This is required in the UAE (United Arab Emirates).

Value space: <0..6>

Range: Select a value between 0 and 6.

Example: `xConfiguration ISDN CliNumType: 0`

xConfiguration ISDN CliNumPlan

Value to be used for calling party number field Plan in the ISDN setup message. If the value is set to 1, the numbering plan in the setup message will signal ISDN. This is required in the UAE (United Arab Emirates).

Value space: <0..14>

Range: Select a value between 0 and 14.

Example: `xConfiguration ISDN CliNumPlan: 0`

xConfiguration ISDN PRI SwitchType

Selects the type of PRI switch the system is connected to.

Value space: <NI/ATT/Euro/Japan>

NI: Select NI if the switch is a National ISDN switch type.

ATT: Select ATT if the switch is an AT&T (for the U.S.) ISDN switch type.

Euro: Select Euro if the switch is an Euro (UK and others) ISDN switch type.

Japan: Select Japan if the switch is a Japanese ISDN switch type.

Example: `xConfiguration ISDN PRI SwitchType: NI`

xConfiguration ISDN PRI InitialRestart

Select if the system should re-initialize the PRI interfaces after the system has been restarted.

Value space: <On/Off>

On: When set to On, the PRI interfaces will be re-initialized after restart of the system.

Off: When set to Off, the PRI interface will not be re-initialized after restart.

Example: `xConfiguration ISDN PRI InitialRestart: On`

xConfiguration ISDN PRI Alert

Select if the system should respond with an alert message to all incoming setup messages, or to the first incoming channel only.

Value space: <On/Off>

On: When set to On, the system will respond with an alert message to all incoming setup messages.

Off: When set to Off (default) the system will respond with an alert message only to the incoming setup message related to the initial channel. Note that if the PBX requires this signalling you will only get connected on the first incoming channel.

Example: `xConfiguration ISDN PRI Alert: Off`

xConfiguration ISDN PRI ChanId

Select if PRI Channel ID information should be sent to the PXB.

Value space: <On/Off>

On: When set to On, the system will signal to the PBX which PRI channels is used for the call.

Note: This setting must be set to On, to work with an Ericsson MD110 PBX.

Off: When set to Off, Channel ID information is not sent. Default setting is Off.

Example: `xConfiguration ISDN PRI ChanId: Off`

xConfiguration ISDN PRI L2WindowSize

Set Layer 2 window size (ISDN Layer 2).

Value space: <1..7>

Range: Select a value between 1 and 7.

Example: `xConfiguration ISDN PRI L2WindowSize: 7`

xConfiguration ISDN PRI NSFTelephony Mode

Configure the NSF (Network Service Facility) mode for telephony.

Value space: <On/Off>

On: Set to On, to enable Network Service Facility for telephony.

Off: Set to Off to disable Network Service Facility for telephony.

Example: `xConfiguration ISDN PRI NSFTelephony Mode: Off`

xConfiguration ISDN PRI NSFTelephony Number

Defines what Service Facility to use for telephony. Requires the NSFTelephony Mode set to On.

Service profiles for AT&T (ref. 1):

- NSF Service
- 0 Disable
- 1 SDN (including GSDN)
- 2 Toll Free Megacom (800)
- 3 Megacom
- 6 ACCUNET Switched Digital Service (including Switched Digital International)
- 7 Long Distance Service (including AT&T World Connect)
- 8 International Toll Free Service (I800)
- 16 AT&T MultiQuest
- 23 Call Redirection Service

Service profiles for Sprint (ref. 2):

- NSF Service
- 0 Reserved
- 1 Private
- 2 Inwatts
- 3 Outwatts
- 4 FX
- 5 TieTrunk

Service profiles for MCI (ref. 3):

- NSF Service
- 1 VNET/Vision
- 2 800
- 3 PRISM1, PRISMII, WATS
- 4 900
- 5 DAL

Value space: <0..31>

Range: Select a value between 0 and 31.

Example: `xConfiguration ISDN PRI NSFTelephony Number: 0`

xConfiguration ISDN PRI NSFVideoTelephony Mode

Configure the NSF (Network Service Facility) mode for video telephony.

Value space: <On/Off>

On: Set to On, to enable Network Service Facility for video telephony.

Off: Set to Off, to disable Network Service Facility for video telephony.

Example: `xConfiguration ISDN PRI NSFVideoTelephony Mode: Off`

xConfiguration ISDN PRI NSFVideoTelephony Number

Defines what Service Facility to use for video telephony. Requires NSFVideoTelephony Mode set to On.

Value space: <0..31>

Range: Select a value between 0 and 31.

Example: `xConfiguration ISDN PRI NSFVideoTelephony Number: 0`

xConfiguration ISDN PRI Interface [1..1] MaxChannels

Set the maximum number of channels the system may use at any given time.

Value space: <1..30>

Range: Select a value between 1 and 30.

Example: `xConfiguration ISDN PRI Interface 1 MaxChannels: 23`

xConfiguration ISDN PRI Interface [1..1] HighChannel

Set the highest numbered B-channel that may be used by the system when selecting channels for outgoing calls.

Value space: <1..31>

Range: Select a value between 1 and 31.

Example: `xConfiguration ISDN PRI Interface 1 HighChannel: 31`

xConfiguration ISDN PRI Interface [1..1] LowChannel

Set the lowest numbered B-channel that may be used by the system when selecting channels for outgoing calls.

Value space: <1..31>

Range: Select a value between 1 and 31.

Example: `xConfiguration ISDN PRI Interface 1 LowChannel: 1`

xConfiguration ISDN PRI Interface [1..1] Search

Selects whether to start searching for available B-channels from the highest numbered channel or from the lowest numbered channel.

Value space: <High/Low>

High: When set to High, the system will start searching for available B-channels from the highest numbered channel.

Low: When set to Low, the system will start searching for available B-channels from the lowest numbered channel.

Example: `xConfiguration ISDN PRI Interface 1 Search: High`

xConfiguration ISDN PRI Interface [1..1] NumberRangeStart

If the PRI line has a range of numbers, the first number in the range should be entered in the Number Range Start field.

Value space: <S: 0, 24>

Format: String with a maximum of 24 characters.

Example: `xConfiguration ISDN PRI Interface 1 NumberRangeStart: ""`

xConfiguration ISDN PRI Interface [1..1] NumberRangeStop

If the PRI line has a range of numbers, the last number in the range is entered in the Number Range Stop field.

Value space: <S: 0, 24>

Format: String with a maximum of 24 characters.

Example: `xConfiguration ISDN PRI Interface 1 NumberRangeStop: ""`

xConfiguration ISDN BRI Alert

Select if the system should respond with an alert message to all incoming setup messages, or to the first incoming channel only.

Value space: <On/Off>

On: When set to On, the system will respond with an alert message to all incoming setup messages.

Off: When set to Off (default) the system will respond with an alert message only to the incoming setup message related to the initial channel. Note that if the PBX requires this signalling you will only get connected on the first incoming channel.

Example: `xConfiguration ISDN BRI Alert: Off`

xConfiguration ISDN BRI ChanId

Select if BRI Channel ID information should be sent to the PxB.

Value space: <On/Off>

On: When set to On, the system will signal to the PBX which BRI channels is used for the call.

Note: This setting must be set to On, to work with an Ericsson MD110 PBX.

Off: When set to Off, Channel ID information is not sent. Default setting is Off.

Example: xConfiguration ISDN BRI ChanId: Off

xConfiguration ISDN BRI InterfaceSearch

Selects whether to start searching for available B-channels from the highest numbered interface or from the lowest numbered interface.

Value space: <High/Low>

High: When set to High, the system will start searching for available B-channels from the highest numbered interface.

Low: When set to Low, the system will start searching for available B-channels from the lowest numbered interface.

Example: xConfiguration ISDN BRI InterfaceSearch: High

xConfiguration ISDN BRI SwitchType

Select the type of BRI switch the system is connected to.

Value space: <NI/ATT/Euro/Japan/1TR6/Australia/FETEX>

NI: Select NI if the switch is a National ISDN switch type.

ATT: Select ATT if the switch is an AT&T (for the U.S.) ISDN switch type.

Euro: Select Euro if the switch is an Euro (UK and others) ISDN switch type.

Japan: Select Japan if the switch is a Japanese ISDN switch type.

1TR6: Select 1TR6 if the switch is an 1TR6 switch type.

Australia: Select Australia if the switch is an Australian ISDN switch type.

FETEX: Select FETEX if the switch is a FETEX ISDN switch type.

Example: xConfiguration ISDN BRI SwitchType: Euro

xConfiguration ISDN BRI AutoActivation

Set the auto activation mode of the BRI interface (ISDN Layer 1).

Value space: <Off/Selected/All>

Off: When set to Off, no interfaces will be activated.

Selected: When set to Selected, the system will activate the interfaces which are already on.

All: When set to All, the system will activate all interfaces regardless of status.

Example: xConfiguration ISDN BRI AutoActivation: All

xConfiguration ISDN BRI MaxDeactiveTime

Set the maximum duration for Layer 1 in de-active state before auto activation (re)starts.

Value space: <1..60>

Range: Select a value between 1 and 60 seconds.

Example: xConfiguration ISDN BRI MaxDeactiveTime: 5

xConfiguration ISDN BRI Interface [1..4] Mode

Select BRI interface 1 to 4, and set the BRI interface mode.

Value space: <On/Off>

On: Set to On to enable the selected BRI interface.

Off: Set to Off to disable the selected BRI interface.

Example: xConfiguration ISDN BRI Interface 1 Mode: Off

xConfiguration ISDN BRI Interface [1..4] DirectoryNumber [1..2] Number

Configure the Directory Numbers for the various BRI interfaces.

Value space: <S: 0, 24>

Format: String with a maximum of 24 characters.

Example: xConfiguration ISDN BRI Interface 1 DirectoryNumber 1 Number: ""

xConfiguration ISDN BRI Interface [1..4] SPID [1..2] Number

Configure the SPID Numbers for the various BRI interfaces.

Value space: <S: 0, 20>

Format: String with a maximum of 20 characters.

Example: xConfiguration ISDN BRI Interface 1 SPID 1 Number: ""

The Network configuration

xConfiguration Network [1..1] Link Network

Set the Ethernet link speed for the network.

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

- Auto:* Autonegotiate link speed.
- 10half:* Force link to 10 Mbps half-duplex.
- 10full:* Force link to 10 Mbps full-duplex.
- 100half:* Force link to 100 Mbps half-duplex.
- 100full:* Force link to 100 Mbps full-duplex.
- 1000full:* Force link to 1 Gbps full-duplex.

Example: xConfiguration Network 1 Link Network: Auto

xConfiguration Network [1..1] Link Endpoint

Set the Ethernet link speed for the endpoint (video system).

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

- Auto:* Autonegotiate link speed.
- 10half:* Force link to 10 Mbps half-duplex.
- 10full:* Force link to 10 Mbps full-duplex.
- 100half:* Force link to 100 Mbps half-duplex.
- 100full:* Force link to 100 Mbps full-duplex.
- 1000full:* Force link to 1 Gbps full-duplex.

Example: xConfiguration Network 1 Link Endpoint: Auto

xConfiguration Network [1..1] Assignment

Define whether to use DHCP or Static IPv4 assignment.

Value space: <Static/DHCP>

- Static:* Set the network assignment to Static and configure the static IPv4 settings (IP Address, SubnetMask and Gateway).
- DHCP:* The system addresses are automatically assigned by the DHCP server.

Example: xConfiguration Network 1 Assignment: DHCP

xConfiguration Network [1..1] DNS Domain Name

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

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

Value space: <S: 0, 64>

Format: String with a maximum of 64 characters.

Example: xConfiguration Network 1 DNS Domain Name: ""

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

Define the network addresses for DNS servers. Up to 3 addresses may be specified. If the network addresses are unknown, contact your administrator or Internet Service Provider.

Value space: <S: 0, 64>

Format: String with a maximum of 64 characters.

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

xConfiguration Network [1..1] IPStack

Select which internet protocols the system will support.

Value space: <IPv4/IPv6>

IPv4: IP version 4 is supported.

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

Example: xConfiguration Network 1 IPStack: IPv4

xConfiguration Network [1..1] IPv4 Address

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

Value space: <S: 0, 64>

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

Example: xConfiguration Network 1 IPv4 Address: "192.0.2.0"

xConfiguration Network [1..1] IPv4 Gateway

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

Value space: <S: 0, 64>

Format: Compact string with a maximum of 64 characters.

Example: xConfiguration Network 1 IPv4 Gateway: "192.0.2.0"

xConfiguration Network [1..1] IPv4 SubnetMask

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

Value space: <S: 0, 64>

Format: Compact string with a maximum of 64 characters.

Example: xConfiguration Network 1 IPv4 SubnetMask: "255.255.255.0"

xConfiguration Network [1..1] IPv6 Address

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

Value space: <S: 0, 64>

Format: The IPv6 address of host name.

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

xConfiguration Network [1..1] IPv6 Gateway

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

Value space: <S: 0, 64>

Format: The IPv6 address of host name.

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

xConfiguration Network [1..1] IPv6 Assignment

Define whether to use Autoconf or Static IPv6 assignment.

Value space: <Static/Autoconf>

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

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

Example: xConfiguration Network 1 IPv6 Assignment: Autoconf

xConfiguration Network [1..1] IPv6 DHCPOptions

Retrieves a set of DHCP options from a DHCPv6 server.

Value space: <On/Off>

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

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

Example: xConfiguration Network 1 IPv6 DHCPOptions: On

xConfiguration Network [1..1] QoS Mode

The QoS (Quality of Service) is a method which handles the priority of audio, video, data, etc. 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.

Value space: <Off/Diffserv>

Off: No QoS method is used.

Diffserv: When you set the QoS Mode to Diffserv you must configure the Diffserv sub menu settings (Audio, Data, Signalling, Video, ICMPv6 and NTP).

Example: xConfiguration Network 1 QoS Mode: Diffserv

xConfiguration Network [1..1] QoS Diffserv Audio

The Diffserv Audio defines which priority Audio packets should have in an IP network. 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.

Value space: <0..63>

Range: Select a value from 0 to 63. A recommended value for Audio packets is Diffserv Code Point (DSCP) AF41, which equals the value 34. If in doubt, contact your network administrator.

Example: xConfiguration Network 1 QoS Diffserv Audio: 0

xConfiguration Network [1..1] QoS Diffserv Data

The Diffserv Data defines which priority Data packets should have in an IP network. 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.

Value space: <0..63>

Range: Select a value from 0 to 63. A recommended value for Data packets is Diffserv Code Point (DSCP) AF23, which equals the value 22. If in doubt, contact your network administrator.

Example: xConfiguration Network 1 QoS Diffserv Data: 0

xConfiguration Network [1..1] QoS Diffserv Signalling

The Diffserv Signalling defines which priority Signalling packets should have in an IP network. 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.

Value space: <0..63>

Range: Select a value from 0 to 63. A recommended value for Signalling packets is Diffserv Code Point (DSCP) AF31, which equals the value 26. If in doubt, contact your network administrator.

Example: xConfiguration Network 1 QoS Diffserv Signalling: 0

xConfiguration Network [1..1] QoS Diffserv ICMPv6

The Diffserv ICMPv6 defines which priority ICMPv6 packets should have in an IP network. 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.

Value space: <0..63>

Range: Select a value from 0 to 63. A recommended value for ICMPv6 packets is Diffserv Code Point (DSCP) CS7, which equals the value 56. If in doubt, contact your network administrator.

Example: `xConfiguration Network 1 QoS Diffserv ICMPv6: 0`

xConfiguration Network [1..1] QoS Diffserv NTP

The Diffserv NTP defines which priority NTP packets should have in an IP network. 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.

Value space: <0..63>

Range: Select a value from 0 to 63. A recommended value for NTP packets is Diffserv Code Point (DSCP) CS7, which equals the value 56. If in doubt, contact your network administrator.

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

xConfiguration Network [1..1] QoS Diffserv Video

The Diffserv Video defines which priority Video packets should have in an IP network. 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.

Value space: <0..63>

Range: Select a value from 0 to 63. A recommended value for Video packets is Diffserv Code Point (DSCP) AF41, which equals the value 34. If in doubt, contact your network administrator.

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

xConfiguration Network [1..1] MTU

Set the Ethernet MTU (Maximum Transmission Unit).

Value space: <576..1500>

Range: Select a value from 576 to 1500 bytes.

Example: `xConfiguration Network 1 MTU: 1500`

xConfiguration Network [1..1] RemoteAccess Allow

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

Value space: <S: 0, 255>

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

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

The NetworkServices configuration

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.

Value space: <Off/Auto/Manual>

Off: The system will not use an NTP server.

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

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

Example: `xConfiguration NetworkServices NTP Mode: Manual`

xConfiguration NetworkServices NTP Address

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

Value space: <S: 0, 64>

Format: String with a maximum of 64 characters.

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

xConfiguration NetworkServices SSH Mode

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

Value space: <On/Off>

On: The SSH protocol is enabled.

Off: The SSH protocol is disabled.

Example: `xConfiguration NetworkServices SSH Mode: On`

The Security configuration

xConfiguration Security Session InactivityTimeout

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

Value space: <0..10000>

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

Example: `xConfiguration Security Session InactivityTimeout: 0`

The SIP configuration

xConfiguration SIP Encryption

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

Value space: <BestEffort/On/Off>

BestEffort: The system will use encryption whenever possible.

1) In Point to point calls: If the far end system supports encryption (AES-128), the call will be encrypted. If not, the call will proceed without encryption.

2) In MultiSite calls: In order to have encrypted MultiSite conferences, all sites must support encryption. If not, the conference will be unencrypted.

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

Off: The system will not use encryption.

Example: `xConfiguration SIP Encryption: BestEffort`

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.

Value space: <S: 0, 255>

Format: Compact string with a maximum of 255 characters.

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

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.

Value space: <S: 0, 128>

Format: String with a maximum of 128 characters.

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

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

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

Value space: <S: 0, 128>

Format: String with a maximum of 128 characters.

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

xConfiguration SIP Profile [1..1] 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.

Value space: <S: 0, 255>

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

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

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

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

Value space: <Auto/Manual>

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

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

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

xConfiguration SIP Profile [1..1] DefaultTransport

Select the transport protocol to be used over the LAN.

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

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

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

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

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

Example: `xConfiguration SIP Profile 1 DefaultTransport: Auto`

xConfiguration SIP Profile [1..1] TlsVerify

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

Value space: <On/Off>

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

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

Example: `xConfiguration SIP Profile 1 TlsVerify: Off`

xConfiguration SIP Profile [1..1] Type

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

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

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

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

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

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

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

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

Broadsoft: To be used when registering to Broadsoft.

Example: `xConfiguration SIP Profile 1 Type: Standard`

xConfiguration SIP Profile [1..1] Outbound

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

Value space: <On/Off>

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

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

Example: `xConfiguration SIP Profile 1 Outbound: Off`

The SystemUnit configuration

xConfiguration SystemUnit Name

Enter the name of the unit. The system unit name will be displayed:

- 1) When acting as an SNMP Agent.
- 2) Towards a DHCP server.

Value space: <s: 0, 50>

Format: String with a maximum of 50 characters.

Example: xConfiguration SystemUnit Name: "ISDN Link"

The T1 configuration

xConfiguration T1 Interface [1..1] CableLength

Specifies the length of the cable connected to the T1 interface.

Value space: <Range1/Range2/Range3/Range4/Range5>

Range1: 0-133 ft (0-40 m)

Range2: 133-266 ft (40-81 m)

Range3: 266-399 ft (81-122 m)

Range4: 399-533 ft (122-162 m)

Range5: 533-655 ft (162-200 m)

Example: xConfiguration T1 Interface 1 CableLength: Range1;

The Time configuration

xConfiguration Time Zone

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

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

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

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

The Experimental configuration

The Experimental settings are for testing only and should not be used unless agreed with Cisco. These settings are not documented and WILL change in later releases.

Chapter 4

Description of the xCommand commands

Description of the xCommands commands

In the following pages you will find a complete list of all xCommand type commands with parameters. We recommend you visit our web site regularly for updated versions of the manual. Go to: ► <http://www.cisco.com/go/isdnlink-docs>

The Boot commands

xCommand Boot

Reboot system.

Requires user role: USER

Example:

```
xCommand Boot
*r BootResult (status=OK):
** end
OK
CUIL reboot request, restarting
Connection closed by foreign host.
```

The Security commands

xCommand Security FIPSMODE Activate

Activate FIPS (140-2) mode. NOTE: Activating FIPS mode implies a reset to factory defaults.

While in FIPS mode, software upgrade is disabled and the following limitations will apply: All calls will be encrypted. IEEE802.1x and SNMP are disabled. The SIP Profile Type setting cannot be Microsoft. The root user is not available (root settings cannot be changed). To exit FIPS mode, perform a factory reset.

Confirm: <Yes>

Requires user role: ADMIN

Parameters:

Confirm(r): <Yes>

Example:

```
xCommand Security FIPSMODE Activate Confirm: "Yes"
OK
*r SecurityFIPSMODEActivateResult (status=OK)
** end
```

The SystemUnit commands

xCommand SystemUnit AdminPassword Set

Set an administrator password to restrict access to the ISDN Link.

Requires user role: USER

Parameters:

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

Example:

```
xCommand SystemUnit AdminPassword Set Password: ""
*r SystemUnitAdminPasswordSetResult (status=OK):
** end
```

xCommand SystemUnit DateTime Set

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

Requires user role: ADMIN

Parameters:

Year: <2008..2037>

Month: <1..12>

Day: <1..31>

Hour: <0..23>

Minute: <0..59>

Second: <0..59>

Example:

```
xCommand SystemUnit DateTime Set Year:2012 Month:7 Day:1 Hour:12 Minute:0
Second:0
*r SystemUnitDateTimeSetResult (status=OK):
** end
```

xCommand SystemUnit DateTime Get

Read the time and date from the system.

Requires user role: USER

Example:

```
xCommand SystemUnit DateTime get
*r SystemUnitDateTimeGetResult (status=OK):
Year: 2012
Month: 7
Day: 1
Hour: 12
Minute: 0
Second: 0
** end
```

xCommand SystemUnit FactoryReset

Reset the codec to factory default settings. The call logs will be deleted and all system parameters will be reset to default values. All files that have been uploaded to the codec will be deleted. The Release key and Option key will not be affected.

As a default the system restarts after the factory reset.

Requires user role: ADMIN

Parameters:

Confirm(r): <Yes>

Example:

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

xCommand SystemUnit SoftwareUpgrade

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

Requires user role: USER

Parameters:

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

UserName: <S: 0, 255>

Password: <S: 0, 255>

Example:

```
xCommand SystemUnit SoftwareUpgrade URL: "ftp://<ftp_server_ip_
address>/<path>/<filename>" UserName: "" Password: ""
*r SystemUnitSoftwareUpgradeResult (status=OK):
** end
```


Chapter 5

Description of the xStatus commands

Description of the xStatus commands

The following pages will list an example of the xStatus commands and the response. Status type commands returns information about the system and system processes. You can query all information or just some of it.

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

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

Call status

ISDN Link operates as a gateway between a Caller and a Callee. Thus xStatus referring to "Caller" is about the calling party or the connection between the calling party and the ISDN Link. Similarly xStatus referring to "Callee" is about the called party or the connection between the ISDN Link and the called party.

xStatus Call

Shows the top level overview of the call status. The call identity is used when querying for additional information about the call.

Example:

```
xStatus Call
*s Call 20 Status: Active
*s Call 20 CallType: Video
*s Call 20 Duration: 2354
*s Call 20 Caller Status: Connected
*s Call 20 Caller Protocol: "SIP"
*s Call 20 Caller RemoteNumber: "firstname.lastname@company.com"
*s Call 20 Caller CallbackNumber: "firstname.lastname@company.com"
*s Call 20 Caller LocalNumber: "isdn-link1@company.com;x-h320number=1234"
*s Call 20 Caller TransmitCallRate: 768
*s Call 20 Caller ReceiveCallRate: 1920
*s Call 20 Caller Encryption Type: "Aes-128"
*s Call 20 Callee Status: Connected
*s Call 20 Callee Protocol: "H320"
*s Call 20 Callee RemoteNumber: "1234"
*s Call 20 Callee CallbackNumber: "1234"
*s Call 20 Callee LocalNumber: "5678"
*s Call 20 Callee TransmitCallRate: 1920
*s Call 20 Callee ReceiveCallRate: 768
*s Call 20 Callee Encryption Type: "None"
** end
```

xStatus Call [1..n] Status

Shows the status of a call. You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<Idle/Initializing/Accepted/Active/Disconnecting>

Example:

```
xStatus Call 20 Status
*s Call 20 Status: Active
** end
```

xStatus Call [1..n] CallType

Shows the call type of the incoming or outgoing call. You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<Video/Audio>

Example:

```
xStatus Call 20 CallType
*s Call 20 CallType: Video
** end
```

xStatus Call [1..n] Duration

Shows the duration of a call (in seconds). You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<Integer>

Example:

```
xStatus Call 20 Duration
*s Call 20 Duration: 2354
** end
```

xStatus Call [1..n] Caller Status

xStatus Call [1..n] Callee Status

Shows the call status between the calling party and the ISDN Link (Caller Status) or between the ISDN Link and the called party (Callee Status). You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<Idle/Dialling/Ringing/Connecting/Connected/Disconnecting/Unknown>

Example:

```
xStatus Call 20 Caller Status
*s Call 20 Caller Status: Connected
** end
```

xStatus Call [1..n] Caller Protocol

xStatus Call [1..n] Callee Protocol

Shows the call protocol being used between the calling party and the ISDN Link (Caller Protocol) or between the ISDN Link and the called party (Callee Protocol). You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<String>

Example:

```
xStatus Call 20 Callee Protocol
*s Call 20 Callee Protocol: "H320"
** end
```

xStatus Call [1..n] Caller RemoteNumber

xStatus Call [1..n] Callee RemoteNumber

Shows the number or URI of the calling party (Caller RemoteNumber) or called party (Callee RemoteNumber). You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<String>

Example:

```
xStatus Call 20 Callee RemoteNumber
*s Call 20 Callee RemoteNumber: "1234"
** end
```

xStatus Call [1..n] Caller CallbackNumber

xStatus Call [1..n] Callee CallbackNumber

Shows the number or URI that ISDN Link shall use when calling back to the calling party (Caller CallbackNumber) or called party (Callee CallbackNumber). You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<String>

Example:

```
xStatus Call 20 Callee CallbackNumber
*s Call 20 Callee CallbackNumber: "1234"
** end
```

xStatus Call [1..n] Caller LocalNumber

Shows the number called to reach the ISDN Link. For SIP callers this would be the number and parameters needed to reach the callee.

Value space of the result returned:

<String>

Example:

```
Calling from a SIP client to an H320 client
-----
xStatus Call 20 Caller LocalNumber
*s Call 20 Caller LocalNumber: "isdn-link1@company.com;x-h320number=1234"
** end
```

```
Calling from an H320 client
-----
xStatus Call 20 Caller LocalNumber
*s Call 20 Caller LocalNumber: "5678"
** end
```

xStatus Call [1..n] Callee LocalNumber

Shows the number from which the ISDN Link is calling. This number is the callback number for the called party. For SIP callee's this would include the number and parameters to perform callback to the caller side.

Value space of the result returned:

<String>

Example:

Calling to an H320 client

```
-----
xStatus Call 20 Callee LocalNumber
*s Call 20 Callee LocalNumber: "5678"
** end
```

Calling to a SIP client from an H320 client

```
-----
xStatus Call 20 Callee LocalNumber
*s Call 20 Callee LocalNumber: "isdn-link1@company.com;x-h320number=1234"
** end
```

xStatus Call [1..n] Caller TransmitCallRate

xStatus Call [1..n] Callee TransmitCallRate

Shows the transmit bandwidth used between the calling party and the ISDN Link (Caller TransmitCallRate) or between the ISDN Link and the called party (Callee TransmitCallRate). The bandwidth is in kilobits per second (kbps). You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<Integer>

Example:

```
xStatus Call 20 Caller TransmitCallRate
*s Call 20 Caller TransmitCallRate: 768
** end
```

xStatus Call [1..n] Caller ReceiveCallRate

xStatus Call [1..n] Callee ReceiveCallRate

Shows the receive bandwidth used between the calling party and the ISDN Link (Caller ReceiveCallRate) or between the ISDN Link and the called party (Callee ReceiveCallRate). The bandwidth is in kilobits per second (kbps). You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<Integer>

Example:

```
xStatus Call 20 Caller ReceiveCallRate
*s Call 20 Caller ReceiveCallRate: 1920
** end
```

xStatus Call [1..n] Caller Encryption Type

xStatus Call [1..n] Callee Encryption Type

Shows the encryption type used between the calling party and the ISDN Link (Caller Encryption Type) or between the ISDN Link and the called party (Callee Encryption Type). You can run the command xStatus Call to find the call identity

Value space of the result returned:

<"None"/"Aes-128">

Example:

```
xStatus Call 20 Caller Encryption Type
*s Call 20 Caller Encryption Type: "Aes-128"
** end
```

ExternalNetwork status

xStatus ExternalNetwork

Shows the top level overview of the External Network interface (V.35, RS-366, RS-449 or RS-530) status.

Cisco TelePresence ISDN Link has three communication interfaces: ISDN BRI, ISDN PRI and External Network. Only one of these interfaces can be enabled at any time.

Example:

```
xStatus ExternalNetwork
*s ExternalNetwork Interface 1 Status: Ready
*s ExternalNetwork Interface 1 ClockRate: 1920
*s ExternalNetwork Interface 1 Channel 1 Status: Disconnected
*s ExternalNetwork Interface 1 Channel 1 Call 10 CallingNumber: "1234"
*s ExternalNetwork Interface 1 Channel 1 Call 10 CauseLocation: 0
*s ExternalNetwork Interface 1 Channel 1 Call 10 ChannelCause: 16
*s ExternalNetwork Interface 1 Channel 1 Call 10 Duration: 771
** end
```

xStatus ExternalNetwork Interface [1..1] Status

Shows the status of the External Network interface.

Ready: The external network is ready to relay calls.

NotReady: The external network is not ready for use, e.g. when clock is missing or the line is broken.

Off: The external network is not enabled (i.e. xConfiguration H320 NetType is not set to External).

Value space of the result returned:

<Ready/NotReady/Off>

Example:

```
xStatus ExternalNetwork Interface 1 Status
*s ExternalNetwork Interface 1 Status: Ready
** end
```

xStatus ExternalNetwork Interface [1..1] ClockRate

Shows the clock rate (in kbps) used for synchronous communication on the external network. This clock is provided by the attached equipment.

Value space of the result returned:

<Integer>

Example:

```
xStatus ExternalNetwork Interface 1 ClockRate
*s ExternalNetwork Interface 1 ClockRate: 1920
** end
```

xStatus ExternalNetwork Interface [1..1] Channel [1..n] Status

Gives information about the call progress in a channel on the external interface. The information indicates the various stages each channel goes through whilst establishing a connection.

Idle: The channel is idle. This state is normally only seen if the channel has never been used.

Calling: The network has acknowledged an incoming call, but the connection is not yet established.

Answering: The network has acknowledged an outgoing call, but the connection is not yet established.

Connect: The connection is established.

Disconnecting: Waiting for the network to confirm the termination of the incoming or outgoing call.

Disconnected: The network has acknowledged the termination of the call. The channel is available for a new call.

Value space of the result returned:

<Idle/Calling/Answering/Connect/Disconnecting/Disconnected>

Example:

```
xStatus ExternalNetwork Interface 1 Channel 1 Status
*s ExternalNetwork Interface 1 Channel 1 Status: Disconnected
** end
```

xStatus ExternalNetwork Interface [1..1] Channel [1..n] Call [1..n] CallingNumber

Shows the number of the system initiating a call on the external network.

Value space of the result returned:

<String>

Example:

```
xStatus ExternalNetwork Interface 1 Channel 1 Call 10 CallingNumber
*s ExternalNetwork Interface 1 Channel 1 Call 10 CallingNumber: "1234"
** end
```

xStatus ExternalNetwork Interface [1..1] Channel [1..n] Call [1..n] CauseLocation

Shows where the disconnection of a call on the external network is originated. The location codes are specified in ITU-T Recommendation Q.931 (the Cause information element).

Value space of the result returned:

<Integer>

Example:

```
xStatus ExternalNetwork Interface 1 Channel 1 Call 10 CauseLocation
*s ExternalNetwork Interface 1 Channel 1 Call 10 CauseLocation: 0
** end
```

xStatus ExternalNetwork Interface [1..1] Channel [1..n] Call [1..n] ChannelCause

Shows the reason why a call on the external network is disconnected. The cause values are specified in ITU-T Recommendation Q.931 (the Cause information element).

Some common cause values are listed below; ITU-T Q.931 contains the complete list.

- 1: Unallocated (unassigned) number
- 2: No route to specified transit network
- 16: Normal call clearing
- 17: User busy
- 18: No user responding
- 21: Call rejected
- 28: Invalid number format
- 29: Facility rejected
- 31: Normal, unspecified
- 34: No circuit/channel available
- 41: Temporary failure
- 58: Bearer capability not presently available
- 65: Bearer capability not implemented
- 69: Requested facility not implemented
- 81: Invalid call reference value
- 88: Incompatible destination
- 100: Invalid information element contents
- 102: Recovery on timer expiry
- 127: Interworking, unspecified
- 128-255: Cisco specific codes

Value space of the result returned:

<Integer>

Example:

```
xStatus ExternalNetwork Interface 1 Channel 1 Call 10 ChannelCause
*s ExternalNetwork Interface 1 Channel 1 Call 10 ChannelCause: 16
** end
```

xStatus ExternalNetwork Interface [1..1] Channel [1..n] Call [1..n] Duration

Shows the duration (in seconds) of a call on the external network.

Value space of the result returned:

<Integer>

Example:

```
xStatus ExternalNetwork Interface 1 Channel 1 Call 10 Duration
*s ExternalNetwork Interface 1 Channel 1 Call 10 Duration: 771
** end
```

H320 status

xStatus H320 NetType

Shows which one of ISDN Link's communication interfaces is enabled. Note that only one interface is enabled at a time.

BRI: The ISDN BRI interface is enabled.

PRI: The ISDN PRI interface is enabled.

External: The external network (NET) interface is enabled.

G703: For testing only.

Value space of the result returned:

<BRI/PRI/External/G703>

Example:

```
xStatus H320 NetType
*s H320 NetType: PRI
** end
```

ISDN status

xStatus ISDN

Shows the top level overview of the ISDN status.

Cisco TelePresence ISDN Link has three communication interfaces: ISDN BRI, ISDN PRI and external network (NET). Only one of these interfaces can be enabled at any time.

Example:

The PRI interface is enabled

```
-----
xStatus ISDN
*s ISDN PRI Interface 1 Status: Ready
*s ISDN PRI Interface 1 BChansFree: 18
*s ISDN PRI Interface 1 BChansTotal: 30
*s ISDN PRI Interface 1 Layer1Status: Synnc
*s ISDN PRI Interface 1 Channel 1 Type: BChannel
*s ISDN PRI Interface 1 Channel 1 Status: Idle
.
.
.
*s ISDN PRI Interface 1 Channel 31 Type: BChannel
*s ISDN PRI Interface 1 Channel 31 Status: Connect
*s ISDN BRI Interface 1 Status: Off
*s ISDN BRI Interface 2 Status: Off
*s ISDN BRI Interface 3 Status: Off
*s ISDN BRI Interface 4 Status: Off
** end
```

The BRI interface is enabled

```
-----
xStatus ISDN
*s ISDN PRI Interface 1 Status: Off
*s ISDN BRI Interface 1 Status: Ready
*s ISDN BRI Interface 1 Channel 1 Type: BChannel
*s ISDN BRI Interface 1 Channel 1 Status: Connect
*s ISDN BRI Interface 1 Channel 1 Call 1 CallingNumber: "1234"
*s ISDN BRI Interface 1 Channel 2 Type: BChannel
*s ISDN BRI Interface 1 Channel 2 Status: Connect
*s ISDN BRI Interface 1 Channel 2 Call 1 CallingNumber: "1234"
*s ISDN BRI Interface 2 Status: Off
*s ISDN BRI Interface 3 Status: Off
```

```
*s ISDN BRI Interface 4 Status: Off
** end
```

Neither the PRI nor the BRI interface is enabled

```
-----
xStatus ISDN
*s ISDN PRI Interface 1 Status: Off
*s ISDN BRI Interface 1 Status: Off
*s ISDN BRI Interface 2 Status: Off
*s ISDN BRI Interface 3 Status: Off
*s ISDN BRI Interface 4 Status: Off
** end
```

xStatus ISDN PRI Interface [1..1] Status

Shows the status of the PRI interface.

Ready: The PRI interface is ready to relay calls.

NotReady: The PRI interface is not ready for use, e.g. when synchronization is missing or the line is broken.

Off: The PRI interface is not enabled (i.e. xConfiguration H320 NetType is not set to PRI).

Value space of the result returned:

<Ready/NotReady/Off>

Example:

```
xStatus ISDN PRI Interface 1 Status
*s ISDN PRI Interface 1 Status: Ready
** end
```

xStatus ISDN PRI Interface [1..1] BChansFree

Shows the number of B channels (bearer channels, 64 kbps) currently available on the PRI interface.

Value space of the result returned:

<0..30>

Example:

```
xStatus ISDN PRI Interface 1 BChansFree
*s ISDN PRI Interface 1 BChansFree: 18
** end
```

xStatus ISDN PRI Interface [1..1] BChansTotal

Shows how many B channels (bearer channels, 64 kbps) the PRI interface offers in total.

Value space of the result returned:

<1..30>

Example:

```
xStatus ISDN PRI Interface 1 BChansTotal
*s ISDN PRI Interface 1 BChansTotal: 30
** end
```

xStatus ISDN PRI Interface [1..1] Layer1Status

Shows whether there are issues relating to framing information (physical layer).

Sync: The system is receiving framing information and the channels are synchronized.

Yellow: The system is receiving framing information, but the remote system is not receiving the transmitted framing information. This may be caused by a broken connector in the TX part of your PRI cable. It could also indicate a weak or noisy signal in the TX part of the PRI cable.

Red: There is no signal and thus the system receives no framing information from the remote site (this has the same effect as pulling out the PRI cable).

Blue: The network on the far side of the CSU (Channel Services Unit) is unavailable. For example, assume that a system is connected via a CSU (Channel Services Unit) as follows: System-CableA-CSU-CableB-Network. If the CSU loses framing/synchronisation from the network, e.g. due to a bad CableB, it will no longer send valid framing out on CableA towards the system. Instead it transmits "Blue Alarm". Seen from the system receiving the blue alarm, this means that the network on the far side of the CSU is unavailable.

Unknown: The system is not able to determine the layer 1 status.

Value space of the result returned:

<Sync/Yellow/Red/Blue/Unknown>

Example:

```
xStatus ISDN PRI Interface 1 Layer1Status
*s ISDN PRI Interface 1 Layer1Status: Sync
** end
```

xStatus ISDN PRI Interface [1..1] Channel [1..n] Type

Shows the channel type for a channel on the PRI interface. The PRI interface comprises a number of channels, each dedicated to either the primary communication or to control and signalling information.

DChannel: The channel is a data channel (D channel), which carries the control and signalling information in ISDN. The bit rate of the D channel on a PRI interface is 64 kbps.

BChannel: The channel is a bearer channel (B channel), which carries the primary video conference or speech call communication in ISDN. The bit rate of a B channel on a PRI interface is 64 kbps.

Value space of the result returned:

<DChannel/BChannel>

Example:

```
xStatus ISDN PRI Interface 1 Channel 1 Type
*s ISDN PRI Interface 1 Channel 1 Type: BChannel
** end
```

xStatus ISDN PRI Interface [1..1] Channel [1..n] Status

Gives information about the call progress in a channel on the PRI interface. The information indicates the various stages each channel goes through whilst establishing a connection.

Idle: The channel is idle. This state is normally only seen if the channel has never been used.

Calling: The network has acknowledged an incoming call, but the connection is not yet established.

Answering: The network has acknowledged an outgoing call, but the connection is not yet established.

Connect: The connection is established.

Disconnecting: Waiting for the network to confirm the termination of the incoming or outgoing call.

Disconnected: The network has acknowledged the termination of the call. The channel is available for a new call.

Value space of the result returned:

<Idle/Calling/Answering/Connect/Disconnecting/Disconnected>

Example:

```
xStatus ISDN PRI Interface 1 Channel 1 Status
*s ISDN PRI Interface 1 Channel 1 Status: Idle
** end
```

xStatus ISDN PRI Interface [1..1] Channel [1..n] Call [1..n] CallingNumber

Shows the number calling into the ISDN PRI interface (for an incoming call), and the number that ISDN Link is making a call to on the ISDN PRI interface (for an outgoing call).

Value space of the result returned:

<String>

Example:

```
xStatus ISDN PRI Interface 1 Channel 1 Call 20 CallingNumber
*s ISDN PRI Interface 1 Channel 1 Call 20 CallingNumber: "1234"
** end
```


xStatus ISDN PRI Interface [1..1] Channel [1..n] Call [1..n] CauseLocation

Shows where the disconnection of a call on the PRI interface is originated. The location codes are specified in ITU-T Recommendation Q.931 (the Cause information element).

Value space of the result returned:

<Integer>

Example:

```
xStatus ISDN PRI Interface 1 Channel 1 Call 20 CauseLocation
*s ISDN PRI Interface 1 Channel 1 Call 20 CauseLocation: 0
** end
```

xStatus ISDN PRI Interface [1..1] Channel [1..n] Call [1..n] ChannelCause

Shows the reason why a call on the PRI interface is disconnected. The cause values are specified in ITU-T Recommendation Q.931 (the Cause information element).

Some common cause values are listed below; ITU-T Q.931 contains the complete list.

- 1: Unallocated (unassigned) number
- 2: No route to specified transit network
- 16: Normal call clearing
- 17: User busy
- 18: No user responding
- 21: Call rejected
- 28: Invalid number format
- 29: Facility rejected
- 31: Normal, unspecified
- 34: No circuit/channel available
- 41: Temporary failure
- 58: Bearer capability not presently available
- 65: Bearer capability not implemented
- 69: Requested facility not implemented
- 81: Invalid call reference value
- 88: Incompatible destination
- 100: Invalid information element contents
- 102: Recovery on timer expiry
- 127: Interworking, unspecified
- 128-255: Cisco specific codes

Value space of the result returned:

<Integer>

Example:

```
xStatus ISDN PRI Interface 1 Channel 1 Call 20 ChannelCause
*s ISDN PRI Interface 1 Channel 1 Call 20 ChannelCause: 16
** end
```

xStatus ISDN PRI Interface [1..1] Channel [1..n] Call [1..n] Duration

Shows the duration (in seconds) of a call on the PRI interface.

Value space of the result returned:

<Integer>

Example:

```
xStatus ISDN PRI Interface 1 Channel 1 Call 20 Duration
*s ISDN PRI Interface 1 Channel 1 Call 20 Duration: 210
** end
```

xStatus ISDN BRI Interface [1..4] Status

Shows the status of the BRI interface.

Ready: The BRI interface is ready to relay calls.

NotReady: The BRI interface is not ready for use, e.g. when synchronization is missing or the line is broken.

Off: The BRI interface is not enabled (i.e. xConfiguration H320 NetType is not set to BRI).

Value space of the result returned:

<Ready/NotReady/Off>

Example:

```
xStatus ISDN BRI Interface 1 Status
*s ISDN BRI Interface 1 Status: Ready
** end
```

xStatus ISDN BRI Interface [1..4] Layer1Alarm

Shows whether or not there are any alarms related to the physical layer, e.g. cable not properly connected to switch.

On: Layer 1 alarm.

Off: No alarm.

Value space of the result returned:

<On/Off>

Example:

```
xStatus ISDN BRI Interface 1 Layer1Alarm
*s ISDN BRI Interface 1 Layer1Alarm: Off
** end
```

xStatus ISDN BRI Interface [1..4] Layer2Alarm

Shows whether there are any alarms related to the data-link layer.
 On: Layer 2 alarm.
 Off: No alarm.

Value space of the result returned:

<On/Off>

Example:

```
xStatus ISDN BRI Interface 1 Layer2Alarm
*s ISDN BRI Interface 1 Layer2Alarm: Off
** end
```

xStatus ISDN BRI Interface [1..4] Channel [1..3] Type

Shows the channel type for a channel on a BRI interface. A BRI interface comprises two channels for the primary communication and one channel for control and signalling information.

DChannel: The channel is a data channel (D channel), which carries the control and signalling information in ISDN. The bit rate of the D channel of a BRI interface is 16 kbps.

BChannel: The channel is a bearer channel (B channel), which carries the primary video conference or speech call communication in ISDN. The bit rate of a B channel on a BRI interface is 64 kbps.

Value space of the result returned:

<DChannel/BChannel>

Example:

```
xStatus ISDN BRI Interface 1 Channel 1 Type
*s ISDN BRI Interface 1 Channel 1 Type: BChannel
** end
```

xStatus ISDN BRI Interface [1..4] Channel [1..3] Status

Gives information about the call progress in a channel on a BRI interface. The information indicates the various stages each channel goes through whilst establishing a connection.

Idle: The channel is idle. This state is normally only seen if the channel has never been used.

Calling: The network has acknowledged an incoming call, but the connection is not yet established.

Answering: The network has acknowledged an outgoing call, but the connection is not yet established.

Connect: The connection is established.

Disconnecting: Waiting for the network to confirm the termination of the incoming or outgoing call.

Disconnected: The network has acknowledged the termination of the call. The channel is available for a new call.

Value space of the result returned:

<Idle/Calling/Answering/Connect/Disconnecting/Disconnected>

Example:

```
xStatus ISDN BRI Interface 1 Channel 1 Status
*s ISDN BRI Interface 1 Channel 1 Status: Connect
** end
```

xStatus ISDN BRI Interface [1..4] Channel [1..3] Call [1..n] CallingNumber

Shows the number calling into the ISDN BRI interface (for an incoming call), and the number that ISDN Link is making a call to on the ISDN BRI interface (for an outgoing call).

Value space of the result returned:

<String>

Example:

```
xStatus ISDN BRI Interface 1 Channel 1 Call 30 CallingNumber
*s ISDN BRI Interface 1 Channel 1 Call 30 CallingNumber: "1234"
** end
```

xStatus ISDN BRI Interface [1..4] Channel [1..3] Call [1..n] CauseLocation

Shows where the disconnection of a call on the BRI interface is originated. The location codes are specified in ITU-T Recommendation Q.931 (the Cause information element).

Value space of the result returned:

<Integer>

Example:

```
xStatus ISDN BRI Interface 1 Channel 1 Call 30 CauseLocation
*s ISDN BRI Interface 1 Channel 1 Call 30 CauseLocation: 0
** end
```

xStatus ISDN BRI Interface [1..4] Channel [1..3] Call [1..n] ChannelCause

Shows the reason why a call on the BRI interface is disconnected. The cause values are specified in ITU-T Recommendation Q.931 (the Cause information element).

Some common cause values are listed below; ITU-T Q.931 contains the complete list.

- 1: Unallocated (unassigned) number
- 2: No route to specified transit network
- 16: Normal clearing
- 17: User busy
- 18: No user responding
- 21: Call rejected
- 28: Invalid number format
- 29: Facility rejected
- 31: Normal, unspecified
- 34: No circuit/channel available
- 41: Temporary failure
- 58: Bearer capability not presently available
- 65: Bearer capability not implemented
- 69: Requested facility not implemented
- 81: Invalid call reference value
- 88: Incompatible destination
- 100: Invalid information element contents
- 102: Recovery on timer expiry
- 127: Interworking, unspecified
- 128-255: Cisco specific codes

Value space of the result returned:

<Integer>

Example:

```
xStatus ISDN BRI Interface 1 Channel 1 Call 30 ChannelCause
*s ISDN BRI Interface 1 Channel 1 Call 30 ChannelCause: 16
** end
```

xStatus ISDN BRI Interface [1..4] Channel [1..3] Call [1..n] Duration

Shows the duration (in seconds) of a call. Applies to calls on a BRI interface channel only.

Value space of the result returned:

<Integer>

Example:

```
xStatus ISDN BRI Interface 1 Channel 1 Call 30 Duration
*s ISDN BRI Interface 1 Channel 1 Call 30 Duration: 270
** end
```

MediaChannels status

ISDN Link operates as a gateway between a Caller and a Callee. Thus xStatus referring to "Caller" is about the calling party or the connection between the calling party and the ISDN Link. Similarly xStatus referring to "Callee" is about the called party or the connection between the ISDN Link and the called party.

xStatus MediaChannels

Shows the top level overview of the media channel status. The example shows the status for an ongoing call. The identities of the call and channels are used when querying additional information.

Example:

```
xStatus MediaChannels
*s MediaChannels Call 20 Caller IncomingAudioChannel 1 Encryption Status:
On
*s MediaChannels Call 20 Caller IncomingAudioChannel 1 Audio Protocol: G722
*s MediaChannels Call 20 Caller IncomingAudioChannel 1 Audio Mute: False
*s MediaChannels Call 20 Caller IncomingAudioChannel 1 Audio Channels: 1
*s MediaChannels Call 20 Caller IncomingAudioChannel 1 Transport RTP Local
IpAddress: "192.0.2.54"
*s MediaChannels Call 20 Caller IncomingAudioChannel 1 Transport RTP Local
Port: 2326
*s MediaChannels Call 20 Caller IncomingAudioChannel 1 Transport RTP Remote
IpAddress: "192.0.2.181"
*s MediaChannels Call 20 Caller IncomingAudioChannel 1 Transport RTP Remote
Port: 2366
*s MediaChannels Call 20 Caller IncomingAudioChannel 1 Transport RTCP Local
IpAddress: "192.0.2.54"
*s MediaChannels Call 20 Caller IncomingAudioChannel 1 Transport RTCP Local
Port: 2327
*s MediaChannels Call 20 Caller IncomingAudioChannel 1 Transport RTCP
Remote IpAddress: "192.0.2.181"
*s MediaChannels Call 20 Caller IncomingAudioChannel 1 Transport RTCP
Remote Port: 2367
*s MediaChannels Call 20 Caller IncomingVideoChannel 2 Encryption Status: On
*s MediaChannels Call 20 Caller IncomingVideoChannel 2 ChannelRole: Main
*s MediaChannels Call 20 Caller IncomingVideoChannel 2 Video Protocol: H264
*s MediaChannels Call 20 Caller IncomingVideoChannel 2 Transport RTP Local
IpAddress: "192.0.2.54"
*s MediaChannels Call 20 Caller IncomingVideoChannel 2 Transport RTP Local
Port: 2342
*s MediaChannels Call 20 Caller IncomingVideoChannel 2 Transport RTP Remote
IpAddress: "192.0.2.181"
*s MediaChannels Call 20 Caller IncomingVideoChannel 2 Transport RTP Remote
Port: 2368
*s MediaChannels Call 20 Caller IncomingVideoChannel 2 Transport RTCP Local
```

```

IpAddress: "192.0.2.54"
*s MediaChannels Call 20 Caller IncomingVideoChannel 2 Transport RTCP Local
Port: 2343
*s MediaChannels Call 20 Caller IncomingVideoChannel 2 Transport RTCP
Remote IpAddress: "192.0.2.181"
*s MediaChannels Call 20 Caller IncomingVideoChannel 2 Transport RTCP
Remote Port: 2369
*s MediaChannels Call 20 Caller IncomingVideoChannel 3 Encryption Status: On
*s MediaChannels Call 20 Caller IncomingVideoChannel 3 ChannelRole:
Presentation
*s MediaChannels Call 20 Caller IncomingVideoChannel 3 Video Protocol: Off
*s MediaChannels Call 20 Caller IncomingVideoChannel 3 Transport RTP Local
IpAddress: "192.0.2.54"
*s MediaChannels Call 20 Caller IncomingVideoChannel 3 Transport RTP Local
Port: 2358
*s MediaChannels Call 20 Caller IncomingVideoChannel 3 Transport RTP Remote
IpAddress: "192.0.2.181"
*s MediaChannels Call 20 Caller IncomingVideoChannel 3 Transport RTP Remote
Port: 2370
*s MediaChannels Call 20 Caller IncomingVideoChannel 3 Transport RTCP Local
IpAddress: "192.0.2.54"
*s MediaChannels Call 20 Caller IncomingVideoChannel 3 Transport RTCP Local
Port: 2359
*s MediaChannels Call 20 Caller IncomingVideoChannel 3 Transport RTCP
Remote IpAddress: "192.0.2.181"
*s MediaChannels Call 20 Caller IncomingVideoChannel 3 Transport RTCP
Remote Port: 2371
*s MediaChannels Call 20 Caller OutgoingAudioChannel 1 Encryption Status: On
*s MediaChannels Call 20 Caller OutgoingAudioChannel 1 Audio Protocol: G722
*s MediaChannels Call 20 Caller OutgoingAudioChannel 1 Audio Channels: 1
*s MediaChannels Call 20 Caller OutgoingAudioChannel 1 Transport RTP Local
IpAddress: "192.0.2.54"
*s MediaChannels Call 20 Caller OutgoingAudioChannel 1 Transport RTP Local
Port: 2326
*s MediaChannels Call 20 Caller OutgoingAudioChannel 1 Transport RTP Remote
IpAddress: "192.0.2.181"
*s MediaChannels Call 20 Caller OutgoingAudioChannel 1 Transport RTP Remote
Port: 2366
*s MediaChannels Call 20 Caller OutgoingAudioChannel 1 Transport RTCP Local
IpAddress: "192.0.2.54"
*s MediaChannels Call 20 Caller OutgoingAudioChannel 1 Transport RTCP Local
Port: 2327
*s MediaChannels Call 20 Caller OutgoingAudioChannel 1 Transport RTCP
Remote IpAddress: "192.0.2.181"
*s MediaChannels Call 20 Caller OutgoingAudioChannel 1 Transport RTCP

Remote Port: 2367
*s MediaChannels Call 20 Caller OutgoingVideoChannel 2 Encryption Status: On
*s MediaChannels Call 20 Caller OutgoingVideoChannel 2 ChannelRole: Main
*s MediaChannels Call 20 Caller OutgoingVideoChannel 2 Video Protocol: H264
*s MediaChannels Call 20 Caller OutgoingVideoChannel 2 Transport RTP Local
IpAddress: "192.0.2.54"
*s MediaChannels Call 20 Caller OutgoingVideoChannel 2 Transport RTP Local
Port: 2342
*s MediaChannels Call 20 Caller OutgoingVideoChannel 2 Transport RTP Remote
IpAddress: "192.0.2.181"
*s MediaChannels Call 20 Caller OutgoingVideoChannel 2 Transport RTP Remote
Port: 2368
*s MediaChannels Call 20 Caller OutgoingVideoChannel 2 Transport RTCP Local
IpAddress: "192.0.2.54"
*s MediaChannels Call 20 Caller OutgoingVideoChannel 2 Transport RTCP Local
Port: 2343
*s MediaChannels Call 20 Caller OutgoingVideoChannel 2 Transport RTCP Remote
IpAddress: "192.0.2.181"
*s MediaChannels Call 20 Caller OutgoingVideoChannel 2 Transport RTCP
Remote Port: 2369
*s MediaChannels Call 20 Caller OutgoingVideoChannel 3 Encryption Status: On
*s MediaChannels Call 20 Caller OutgoingVideoChannel 3 ChannelRole:
Presentation
*s MediaChannels Call 20 Caller OutgoingVideoChannel 3 Video Protocol: Off
*s MediaChannels Call 20 Caller OutgoingVideoChannel 3 Transport RTP Local
IpAddress: "192.0.2.54"
*s MediaChannels Call 20 Caller OutgoingVideoChannel 3 Transport RTP Local
Port: 2358
*s MediaChannels Call 20 Caller OutgoingVideoChannel 3 Transport RTP Remote
IpAddress: "192.0.2.181"
*s MediaChannels Call 20 Caller OutgoingVideoChannel 3 Transport RTP Remote
Port: 2370
*s MediaChannels Call 20 Caller OutgoingVideoChannel 3 Transport RTCP Local
IpAddress: "192.0.2.54"
*s MediaChannels Call 20 Caller OutgoingVideoChannel 3 Transport RTCP Local
Port: 2359
*s MediaChannels Call 20 Caller OutgoingVideoChannel 3 Transport RTCP Remote
IpAddress: "192.0.2.181"
*s MediaChannels Call 20 Caller OutgoingVideoChannel 3 Transport RTCP
Remote Port: 2371
*s MediaChannels Call 20 Callee IncomingAudioChannel 1 Encryption Status:

```

```

Off
*s MediaChannels Call 20 Callee IncomingAudioChannel 1 Audio Protocol: G722
*s MediaChannels Call 20 Callee IncomingAudioChannel 1 Audio Mute: False
*s MediaChannels Call 20 Callee IncomingAudioChannel 1 Audio Channels: 1
*s MediaChannels Call 20 Callee IncomingVideoChannel 2 Encryption Status:
Off
*s MediaChannels Call 20 Callee IncomingVideoChannel 2 ChannelRole: Main
*s MediaChannels Call 20 Callee IncomingVideoChannel 2 Video Protocol: H264
*s MediaChannels Call 20 Callee IncomingVideoChannel 3 Encryption Status:
Off
*s MediaChannels Call 20 Callee IncomingVideoChannel 3 ChannelRole:
Presentation
*s MediaChannels Call 20 Callee IncomingVideoChannel 3 Video Protocol: Off
*s MediaChannels Call 20 Callee OutgoingAudioChannel 1 Encryption Status:
Off
*s MediaChannels Call 20 Callee OutgoingAudioChannel 1 Audio Protocol: G722
*s MediaChannels Call 20 Callee OutgoingAudioChannel 1 Audio Channels: 1
*s MediaChannels Call 20 Callee OutgoingVideoChannel 2 Encryption Status:
Off
*s MediaChannels Call 20 Callee OutgoingVideoChannel 2 ChannelRole: Main
*s MediaChannels Call 20 Callee OutgoingVideoChannel 2 Video Protocol: H264
*s MediaChannels Call 20 Callee OutgoingVideoChannel 3 Encryption Status:
Off
*s MediaChannels Call 20 Callee OutgoingVideoChannel 3 ChannelRole:
Presentation
*s MediaChannels Call 20 Callee OutgoingVideoChannel 3 Video Protocol: Off
** end

```

xStatus MediaChannels Call [1..n] Caller IncomingAudioChannel [1..n] Encryption Status
xStatus MediaChannels Call [1..n] Caller OutgoingAudioChannel [1..n] Encryption Status
xStatus MediaChannels Call [1..n] Callee IncomingAudioChannel [1..n] Encryption Status
xStatus MediaChannels Call [1..n] Callee OutgoingAudioChannel [1..n] Encryption Status

Shows the encryption status on the incoming or outgoing audio channel, on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

Value space of the result returned:

<On/Off>

Example:

```

xStatus MediaChannels Call 20 IncomingAudioChannel 1 Encryption Status
*s MediaChannels Call 20 IncomingAudioChannel 1 Encryption Status: On
** end

```

xStatus MediaChannels Call [1..n] Caller IncomingAudioChannel [1..n] Audio Protocol
xStatus MediaChannels Call [1..n] Caller OutgoingAudioChannel [1..n] Audio Protocol
xStatus MediaChannels Call [1..n] Callee IncomingAudioChannel [1..n] Audio Protocol
xStatus MediaChannels Call [1..n] Callee OutgoingAudioChannel [1..n] Audio Protocol

Shows the audio algorithm for the incoming or outgoing audio channel, on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

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

G7221: The G.722.1 algorithm is a licensed royalty-free ITU-T standard.

G711Mu: The G.711 Mu-law compression algorithm is used in North America and Japan.

G711A: The G.711 A-law compression algorithm is used in Europe and the rest of the world.

Value space of the result returned:

<G722/G7221/G711Mu/G711A>

Example:

```

xStatus MediaChannels Call 20 IncomingAudioChannel 1 Audio Protocol
*s MediaChannels Call 20 IncomingAudioChannel 1 Audio Protocol: G722
** end

```

xStatus MediaChannels Call [1..n] Caller IncomingAudioChannel [1..n] Audio Mute
xStatus MediaChannels Call [1..n] Callee IncomingAudioChannel [1..n] Audio Mute

Audio mute status of incoming audio on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

True: Audio is muted.

False: Audio is not muted.

Value space of the result returned:

<True/False>

Example:

```

xStatus MediaChannels Call 20 IncomingAudioChannel 1 Audio Mute
*s MediaChannels Call 20 IncomingAudioChannel 1 Audio Mute: False
** end

```

xStatus MediaChannels Call [1..n] Caller IncomingAudioChannel [1..n] Audio Channels
xStatus MediaChannels Call [1..n] Caller OutgoingAudioChannel [1..n] Audio Channels
xStatus MediaChannels Call [1..n] Callee IncomingAudioChannel [1..n] Audio Channels
xStatus MediaChannels Call [1..n] Callee OutgoingAudioChannel [1..n] Audio Channels

Shows the number of incoming or outgoing audio channels, on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 20 IncomingAudioChannel 1 Audio Channels
*s MediaChannels Call 20 IncomingAudioChannel 1 Audio Channels: 1
** end
```

xStatus MediaChannels Call [1..n] Caller IncomingAudioChannel [1..n] Transport RTP Local IpAddress

xStatus MediaChannels Call [1..n] Caller OutgoingAudioChannel [1..n] Transport RTP Local IpAddress

xStatus MediaChannels Call [1..n] Callee IncomingAudioChannel [1..n] Transport RTP Local IpAddress

xStatus MediaChannels Call [1..n] Callee OutgoingAudioChannel [1..n] Transport RTP Local IpAddress

Shows the local IP address of the Real-time Transport Protocol (RTP) port for the incoming or outgoing audio, on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 20 IncomingAudioChannel 1 Transport RTP Local
IpAddress
*s MediaChannels Call 20 IncomingAudioChannel 1 Transport RTP Local
IpAddress: "192.0.2.54"
** end
```

xStatus MediaChannels Call [1..n] Caller IncomingAudioChannel [1..n] Transport RTP Local Port

xStatus MediaChannels Call [1..n] Caller OutgoingAudioChannel [1..n] Transport RTP Local Port

xStatus MediaChannels Call [1..n] Callee IncomingAudioChannel [1..n] Transport RTP Local Port

xStatus MediaChannels Call [1..n] Callee OutgoingAudioChannel [1..n] Transport RTP Local Port

Shows the local UDP port number of the Real-time Transport Protocol (RTP) port for the incoming or outgoing audio, on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 20 IncomingAudioChannel 1 Transport RTP Local
Port
*s MediaChannels Call 20 IncomingAudioChannel 1 Transport RTP Local Port:
2326
** end
```

xStatus MediaChannels Call [1..n] Caller IncomingAudioChannel [1..n] Transport RTP Remote IpAddress

xStatus MediaChannels Call [1..n] Caller OutgoingAudioChannel [1..n] Transport RTP Remote IpAddress

xStatus MediaChannels Call [1..n] Callee IncomingAudioChannel [1..n] Transport RTP Remote IpAddress

xStatus MediaChannels Call [1..n] Callee OutgoingAudioChannel [1..n] Transport RTP Remote IpAddress

Shows the remote IP address of the Real-time Transport Protocol (RTP) port for the incoming or outgoing audio, on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 20 IncomingAudioChannel 1 Transport RTP Remote
IpAddress
*s MediaChannels Call 20 IncomingAudioChannel 1 Transport RTP Remote
IpAddress: "192.0.2.181"
** end
```

xStatus MediaChannels Call [1..n] Caller IncomingAudioChannel [1..n] Transport RTP Remote Port

xStatus MediaChannels Call [1..n] Caller OutgoingAudioChannel [1..n] Transport RTP Remote Port

xStatus MediaChannels Call [1..n] Callee IncomingAudioChannel [1..n] Transport RTP Remote Port

xStatus MediaChannels Call [1..n] Callee OutgoingAudioChannel [1..n] Transport RTP Remote Port

Shows the remote UDP port number of the Real-time Transport Protocol (RTP) port for the incoming or outgoing audio, on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 20 IncomingAudioChannel 1 Transport RTP Remote
Port
*s MediaChannels Call 20 IncomingAudioChannel 1 Transport RTP Remote Port:
2366
** end
```

xStatus MediaChannels Call [1..n] Caller IncomingAudioChannel [1..n] Transport RTCP Local IpAddress

xStatus MediaChannels Call [1..n] Caller OutgoingAudioChannel [1..n] Transport RTCP Local IpAddress

xStatus MediaChannels Call [1..n] Callee IncomingAudioChannel [1..n] Transport RTCP Local IpAddress

xStatus MediaChannels Call [1..n] Callee OutgoingAudioChannel [1..n] Transport RTCP Local IpAddress

Shows the local IP address of the Real-time Transport Control Protocol (RTCP) port for the incoming or outgoing audio, on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 20 IncomingAudioChannel 1 Transport RTCP Local
IpAddress
*s MediaChannels Call 20 IncomingAudioChannel 1 Transport RTCP Local
IpAddress: "192.0.2.54"
** end
```

xStatus MediaChannels Call [1..n] Caller IncomingAudioChannel [1..n] Transport RTCP Local Port

xStatus MediaChannels Call [1..n] Caller OutgoingAudioChannel [1..n] Transport RTCP Local Port

xStatus MediaChannels Call [1..n] Callee IncomingAudioChannel [1..n] Transport RTCP Local Port

xStatus MediaChannels Call [1..n] Callee OutgoingAudioChannel [1..n] Transport RTCP Local Port

Shows the local UDP port number of the Real-time Transport Control Protocol (RTCP) port for the incoming or outgoing audio, on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 20 IncomingAudioChannel 1 Transport RTCP Local
Port
*s MediaChannels Call 20 IncomingAudioChannel 1 Transport RTCP Local Port:
2327
** end
```

xStatus MediaChannels Call [1..n] Caller IncomingAudioChannel [1..n] Transport RTCP Remote IpAddress

xStatus MediaChannels Call [1..n] Caller OutgoingAudioChannel [1..n] Transport RTCP Remote IpAddress

xStatus MediaChannels Call [1..n] Callee IncomingAudioChannel [1..n] Transport RTCP Remote IpAddress

xStatus MediaChannels Call [1..n] Callee OutgoingAudioChannel [1..n] Transport RTCP Remote IpAddress

Shows the remote IP address of the Real-time Transport Control Protocol (RTCP) port for the incoming or outgoing audio, on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 20 IncomingAudioChannel 1 Transport RTCP Remote
IpAddress
*s MediaChannels Call 20 IncomingAudioChannel 1 Transport RTCP Remote
IpAddress: "192.0.2.181"
** end
```


xStatus MediaChannels Call [1..n] Caller IncomingAudioChannel [1..n] Transport RTCP Remote Port

xStatus MediaChannels Call [1..n] Caller OutgoingAudioChannel [1..n] Transport RTCP Remote Port

xStatus MediaChannels Call [1..n] Callee IncomingAudioChannel [1..n] Transport RTCP Remote Port

xStatus MediaChannels Call [1..n] Callee OutgoingAudioChannel [1..n] Transport RTCP Remote Port

Shows the remote UDP port number of the Real-time Transport Control Protocol (RTCP) port for the incoming or outgoing audio, on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 20 IncomingAudioChannel 1 Transport RTCP Remote Port
*s MediaChannels Call 20 IncomingAudioChannel 1 Transport RTCP Remote Port: 2367
** end
```

xStatus MediaChannels Call [1..n] Caller IncomingVideoChannel [1..n] Encryption Status

xStatus MediaChannels Call [1..n] Caller OutgoingVideoChannel [1..n] Encryption Status

xStatus MediaChannels Call [1..n] Callee IncomingVideoChannel [1..n] Encryption Status

xStatus MediaChannels Call [1..n] Callee OutgoingVideoChannel [1..n] Encryption Status

Shows the encryption status on the incoming or outgoing video channel, on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

Value space of the result returned:

<On/Off>

Example:

```
xStatus MediaChannels Call 20 IncomingVideoChannel 2 Encryption Status
*s MediaChannels Call 20 IncomingVideoChannel 2 Encryption Status: On
** end
```

xStatus MediaChannels Call [1..n] Caller IncomingVideoChannel [1..n] ChannelRole

xStatus MediaChannels Call [1..n] Caller OutgoingVideoChannel [1..n] ChannelRole

xStatus MediaChannels Call [1..n] Callee IncomingVideoChannel [1..n] ChannelRole

xStatus MediaChannels Call [1..n] Callee OutgoingVideoChannel [1..n] ChannelRole

Shows whether the incoming or outgoing channel is the main video channel or the presentation channel. The status is for the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee)

Value space of the result returned:

<Main/Presentation/Legacy>

Example:

```
xStatus MediaChannels Call 20 IncomingVideoChannel 2 ChannelRole
*s MediaChannels Call 20 IncomingVideoChannel 2 ChannelRole: Main
** end
```

xStatus MediaChannels Call [1..n] Caller IncomingVideoChannel [1..n] Video Protocol

xStatus MediaChannels Call [1..n] Caller OutgoingVideoChannel [1..n] Video Protocol

xStatus MediaChannels Call [1..n] Callee IncomingVideoChannel [1..n] Video Protocol

xStatus MediaChannels Call [1..n] Callee OutgoingVideoChannel [1..n] Video Protocol

Shows the video algorithm for the incoming or outgoing video channel, on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

H264: The H.264 algorithm is an ITU-T standard for video compression.

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

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

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

Value space of the result returned:

<H264/H263pp/H263/H261>

Example:

```
xStatus MediaChannels Call 20 IncomingVideoChannel 2 Video Protocol
*s MediaChannels Call 20 IncomingVideoChannel 2 Video Protocol: H264
** end
```


xStatus MediaChannels Call [1..n] Caller IncomingVideoChannel [1..n] Transport RTP Local IpAddress

xStatus MediaChannels Call [1..n] Caller OutgoingVideoChannel [1..n] Transport RTP Local IpAddress

xStatus MediaChannels Call [1..n] Callee IncomingVideoChannel [1..n] Transport RTP Local IpAddress

xStatus MediaChannels Call [1..n] Callee OutgoingVideoChannel [1..n] Transport RTP Local IpAddress

Shows the local IP address of the Real-time Transport Protocol (RTP) port for the incoming or outgoing video, on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 20 IncomingVideoChannel 2 Transport RTP Local
IpAddress
*s MediaChannels Call 20 IncomingVideoChannel 2 Transport RTP Local
IpAddress: "192.0.2.54"
** end
```

xStatus MediaChannels Call [1..n] Caller IncomingVideoChannel [1..n] Transport RTP Local Port

xStatus MediaChannels Call [1..n] Caller OutgoingVideoChannel [1..n] Transport RTP Local Port

xStatus MediaChannels Call [1..n] Callee IncomingVideoChannel [1..n] Transport RTP Local Port

xStatus MediaChannels Call [1..n] Callee OutgoingVideoChannel [1..n] Transport RTP Local Port

Shows the local UDP port number of the Real-time Transport Protocol (RTP) port for the incoming or outgoing video, on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 20 IncomingVideoChannel 2 Transport RTP Local
Port
*s MediaChannels Call 20 IncomingVideoChannel 2 Transport RTP Local Port:
2342
** end
```

xStatus MediaChannels Call [1..n] Caller IncomingVideoChannel [1..n] Transport RTP Remote IpAddress

xStatus MediaChannels Call [1..n] Caller OutgoingVideoChannel [1..n] Transport RTP Remote IpAddress

xStatus MediaChannels Call [1..n] Callee IncomingVideoChannel [1..n] Transport RTP Remote IpAddress

xStatus MediaChannels Call [1..n] Callee OutgoingVideoChannel [1..n] Transport RTP Remote IpAddress

Shows the remote IP address of the Real-time Transport Protocol (RTP) port for the incoming or outgoing video, on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 20 IncomingVideoChannel 2 Transport RTP Remote
IpAddress
*s MediaChannels Call 20 IncomingVideoChannel 2 Transport RTP Remote
IpAddress: "192.0.2.181"
** end
```

xStatus MediaChannels Call [1..n] Caller IncomingVideoChannel [1..n] Transport RTP Remote Port

xStatus MediaChannels Call [1..n] Caller OutgoingVideoChannel [1..n] Transport RTP Remote Port

xStatus MediaChannels Call [1..n] Callee IncomingVideoChannel [1..n] Transport RTP Remote Port

xStatus MediaChannels Call [1..n] Callee OutgoingVideoChannel [1..n] Transport RTP Remote Port

Shows the remote UDP port number of the Real-time Transport Protocol (RTP) port for the incoming or outgoing video, on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 20 IncomingVideoChannel 2 Transport RTP Remote
Port
*s MediaChannels Call 20 IncomingVideoChannel 2 Transport RTP Remote Port:
2368
** end
```

xStatus MediaChannels Call [1..n] Caller IncomingVideoChannel [1..n] Transport RTCP Local IpAddress

xStatus MediaChannels Call [1..n] Caller OutgoingVideoChannel [1..n] Transport RTCP Local IpAddress

xStatus MediaChannels Call [1..n] Callee IncomingVideoChannel [1..n] Transport RTCP Local IpAddress

xStatus MediaChannels Call [1..n] Callee OutgoingVideoChannel [1..n] Transport RTCP Local IpAddress

Shows the local IP address of the Real-time Transport Control Protocol (RTCP) port for the incoming or outgoing video, on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 20 IncomingVideoChannel 2 Transport RTCP Local
IpAddress
*s MediaChannels Call 20 IncomingVideoChannel 2 Transport RTCP Local
IpAddress: "192.0.2.54"
** end
```

xStatus MediaChannels Call [1..n] Caller IncomingVideoChannel [1..n] Transport RTCP Local Port

xStatus MediaChannels Call [1..n] Caller OutgoingVideoChannel [1..n] Transport RTCP Local Port

xStatus MediaChannels Call [1..n] Callee IncomingVideoChannel [1..n] Transport RTCP Local Port

xStatus MediaChannels Call [1..n] Callee OutgoingVideoChannel [1..n] Transport RTCP Local Port

Shows the local UDP port number of the Real-time Transport Control Protocol (RTCP) port for the incoming or outgoing video, on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 20 IncomingVideoChannel 2 Transport RTCP Local
Port
*s MediaChannels Call 20 IncomingVideoChannel 2 Transport RTCP Local Port:
2343
** end
```

xStatus MediaChannels Call [1..n] Caller IncomingVideoChannel [1..n] Transport RTCP Remote IpAddress

xStatus MediaChannels Call [1..n] Caller OutgoingVideoChannel [1..n] Transport RTCP Remote IpAddress

xStatus MediaChannels Call [1..n] Callee IncomingVideoChannel [1..n] Transport RTCP Remote IpAddress

xStatus MediaChannels Call [1..n] Callee OutgoingVideoChannel [1..n] Transport RTCP Remote IpAddress

Shows the remote IP address of the Real-time Transport Control Protocol (RTCP) port for the incoming or outgoing video, on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 20 IncomingVideoChannel 2 Transport RTCP Remote
IpAddress
*s MediaChannels Call 20 IncomingVideoChannel 2 Transport RTCP Remote
IpAddress: "192.0.2.181"
** end
```

xStatus MediaChannels Call [1..n] Caller IncomingVideoChannel [1..n] Transport RCTP Remote Port

xStatus MediaChannels Call [1..n] Caller OutgoingVideoChannel [1..n] Transport RTCP Remote Port

xStatus MediaChannels Call [1..n] Callee IncomingVideoChannel [1..n] Transport RTCP Remote Port

xStatus MediaChannels Call [1..n] Callee OutgoingVideoChannel [1..n] Transport RTCP Remote Port

Shows the remote UDP port number of the Real-time Transport Control Protocol (RTCP) port for the incoming or outgoing video, on the connection between the ISDN Link and the calling party (Caller) or between the ISDN Link and the called party (Callee).

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 20 IncomingVideoChannel 2 Transport RTCP Remote
Port
*s MediaChannels Call 20 IncomingVideoChannel 2 Transport RTCP Remote Port:
2369
** end
```

Network status

xStatus Network

Shows the top level overview of the network status.

Example:

```
xStatus Network
*s Network 1 Ethernet MacAddress: "00:50:60:70:80:90"
*s Network 1 Ethernet Link Network: "No LAN"
*s Network 1 Ethernet Link Endpoint: "1000full"
*s Network 1 IPv4 Address: "192.0.2.149"
*s Network 1 IPv4 SubnetMask: "255.255.255.0"
*s Network 1 IPv4 Gateway: "192.0.2.10"
*s Network 1 IPv4 DNS Domain Name: "www.example.com www.example.int"
*s Network 1 IPv4 DNS Server 1 Address: "192.0.2.60"
*s Network 1 IPv4 DNS Server 2 Address: "192.0.2.61"
*s Network 1 IPv4 DNS Server 3 Address: ""
*s Network 1 IPv6 Address: "2001:DB8::1234"
*s Network 1 IPv6 Gateway: "2001:DB8::0001"
*s Network 1 MTU: 1500
** end
```

xStatus Network [1..1] Ethernet MacAddress

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

Value space of the result returned:

<String>

Example:

```
xStatus Network 1 Ethernet MacAddress
*s Network 1 Ethernet MacAddress: "00:50:60:70:80:90"
** end
```

xStatus Network [1..1] Ethernet Link Network

Shows the Ethernet link status for the network Ethernet port. This port is used if the ISDN Link is connected to a network (LAN).

Value space of the result returned:

<"No LAN"/"10half"/"10full"/"100half"/"100full"/"1000full">

Example:

```
xStatus Network 1 Ethernet Link Network
*s Network 1 Ethernet Link Network: "No LAN"
** end
```

xStatus Network [1..1] Ethernet Link Endpoint

Shows the Ethernet link status for the endpoint Ethernet port. This port is used to connect an endpoint (codec) to the ISDN Link

Value space of the result returned:

<"No LAN"/"10half"/"10full"/"100half"/"100full"/"1000full">

Example:

```
xStatus Network 1 Ethernet Link Endpoint
*s Network 1 Ethernet Link Endpoint: "1000full"
** end
```

xStatus Network [1..1] IPv4 Address

Shows the IPv4 address that uniquely identifies this system.

Value space of the result returned:

<String>

Example:

```
xStatus Network 1 IPv4 Address
*s Network 1 IPv4 Address: "192.0.2.149"
** end
```

xStatus Network [1..1] IPv4 SubnetMask

Shows the subnet mask which determines which subnet an IPv4 address belongs to.

Value space of the result returned:

<String>

Example:

```
xStatus Network 1 IPv4 SubnetMask
*s Network 1 IPv4 SubnetMask: "255.255.255.0"
** end
```

xStatus Network [1..1] IPv4 Gateway

Shows the address of the IPv4 gateway.

Value space of the result returned:

<String>

Example:

```
xStatus Network 1 IPv4 Gateway
*s Network 1 IPv4 Gateway: "192.0.2.10"
** end
```

xStatus Network [1..1] IPv4 DNS Domain Name

Shows the domain name.

Value space of the result returned:

<String>

Example:

```
xStatus Network 1 IPv4 DNS Domain Name
*s Network 1 IPv4 DNS Domain Name: "www.example.com www.example.int"
** end
```

xStatus Network [1..1] IPv4 DNS Server [1..3] Address

Shows the IP address of the DNS server.

Value space of the result returned:

<String>

Example:

```
xStatus Network 1 IPv4 DNS Server 1. Address
*s Network 1 IPv4 DNS Server 1 Address: "192.0.2.60"
** end
```

xStatus Network [1..1] IPv6 Address

Shows the IPv6 address that uniquely identifies this system.

Value space of the result returned:

<String>

Example:

```
xStatus Network 1 IPv6 Address
*s Network 1 IPv6 Address: "2001:DB8::1234"
** end
```

xStatus Network [1..1] IPv6 Gateway

Shows the address of the IPv6 gateway.

Value space of the result returned:

<String>

Example:

```
xStatus Network 1 IPv6 Gateway
*s Network 1 IPv6 Gateway: "2001:DB8::0001"
** end
```

xStatus Network [1..1] MTU

Shows the MTU (Maximum Transmission Unit) size for the network.

Value space of the result returned:

<Integer>

Example:

```
xStatus Network 1 MTU
*s Network 1 MTU: 1500
** end
```

SIP status

xStatus SIP

Shows the top level overview of the SIP status.

Example:

```
xStatus SIP
*s SIP Profile 1 Proxy 1 Status: Active
*s SIP Profile 1 Proxy 1 Address: "192.0.1.50"
*s SIP Profile 1 Secure: True
*s SIP Profile 1 Verified: False
*s SIP Profile 1 Authentication: Off
*s SIP Profile 1 Registration 1 Status: Registered
*s SIP Profile 1 Registration 1 Reason: ""
*s SIP Profile 1 Registration 1 URI: "isdn-link1@company.com"
** end
```

xStatus SIP Profile [1..1] Proxy [1..1] Status

Shows the status of the communication between the endpoint and the SIP Proxy server.

Active: The communication between the endpoint and the SIP Proxy is active.

DNSFailed: The attempt to establish communication to the DNS server failed.

Off: There is no communication between the endpoint and the SIP Proxy.

Timeout: The attempt to establish communication to the SIP Proxy timed out.

UnableTCP: The system is unable to use TCP as the transport method.

UnableTLS: The system is unable to use TLS as the transport method.

Unknown: The status of the communication is not known.

Value space of the result returned:

<Active/DNSFailed/Off/Timeout/UnableTCP/UnableTLS/Unknown>

Example:

```
xStatus SIP Profile 1 Proxy 1 Status
*s SIP Profile 1 Proxy 1 Status: Active
** end
```

xStatus SIP Profile [1..1] Proxy [1..1] Address

Shows the address of the SIP Proxy that the system communicates with.

Value space of the result returned:

<String>

Example:

```
xStatus SIP Profile 1 Proxy 1 Address
*s SIP Profile 1 Proxy 1 Address: "192.0.2.50"
** end
```

xStatus SIP Profile [1..1] Secure

Shows the encryption status of the signalling with the SIP Proxy server.

Value space of the result returned:

<True/False>

Example:

```
xStatus SIP Profile 1 Secure
*s SIP Profile 1 Secure: True
** end
```

xStatus SIP Profile [1..1] Verified

Shows whether or not the SSL certificate of the server that the video system / codec tries to register to is included in the codec's trusted CA-list. The server will typically be a Cisco VCS or Cisco Unified CM (CUCM).

True: The server's SIP certificate is checked against the trusted CA-list on the codec and found valid. Additionally, the fully qualified domain name of the server matches the valid certificate.

False: A TLS connection is not set up because the SIP certificate verification failed or the domain name did not match. Note that the status will also return False when TLS is not used (SIP Profile [1..1] DefaultTransport not set to TLS) or certificate verification is switched off (SIP Profile TlsVerify set to Off).

Value space of the result returned:

<True/False>

Example:

```
xStatus SIP Profile 1 Verified
*s SIP Profile 1 Verified: False
** end
```

xStatus SIP Profile [1..1] Authentication

Shows which authentication mechanism is used when registering to the SIP Proxy Server.
 Digest: Uses the Digest access authentication method, as specified by RFC 2069.
 NTLM: Uses the NTLM authentication method, which is a Microsoft authentication protocol.
 Off: No authentication mechanism is used.

Value space of the result returned:

<Digest/NTLM/Off>

Example:

```
xStatus SIP Profile 1 Authentication
*s SIP Profile 1 Authentication: Off
** end
```

xStatus SIP Profile [1..1] Registration [1..n] Status

Shows the status of the registration to the SIP Proxy Server.
 Deregister: The system is in the process of de-registering to the SIP Proxy.
 Failed: The system failed to register to the SIP Proxy.
 Inactive: The system is not registered to any SIP Proxy.
 Registered: The system is registered to the SIP Proxy.
 Registering: The system is in the process of registering to the SIP Proxy.

Value space of the result returned:

<Deregister/Failed/Inactive/Registered/Registering>

Example:

```
xStatus SIP Profile 1 Registration 1 Status
*s SIP Profile 1 Registration 1 Status: Registered
** end
```

xStatus SIP Profile [1..1] Registration [1..n] Reason

Shows a message to explain the reason why the SIP registration failed.

Value space of the result returned:

<String>

Example:

```
xStatus SIP Profile 1 Registration 1 Reason
*s SIP Profile 1 Registration 1 Reason: ""
** end
```

xStatus SIP Profile [1..1] Registration [1..n] URI

Shows the URI used for registration to the SIP Proxy server.

Value space of the result returned:

<String>

Example:

```
xStatus SIP Profile 1 Registration 1 URI
*s SIP Profile 1 Registration 1 URI: "isdn-link1@company.com"
** end
```

SystemUnit status

xStatus SystemUnit

Shows the top level overview of the system unit status.

Example:

```
xStatus SystemUnit
*s SystemUnit ProductType: "Cisco TelePresence ISDN link"
*s SystemUnit Uptime: 2158
*s SystemUnit Software Version: "IL1.0.0"
*s SystemUnit Software Name: "s51500"
*s SystemUnit Software ReleaseDate: "2012-05-30"
*s SystemUnit Software BootStage1: "U-boot Stage1 2010.06-4"
*s SystemUnit Software BootStage2: "U-boot Stage2 2010.06-8"
*s SystemUnit Software MaxVideoCalls: 3
*s SystemUnit Software MaxAudioCalls: 3
*s SystemUnit Hardware Module SerialNumber: "FTT1234567V"
*s SystemUnit Hardware Module Identifier: "01"
*s SystemUnit Hardware MainBoard SerialNumber: "F0999999"
*s SystemUnit Hardware MainBoard Identifier: "102030-1 [00]"
*s SystemUnit Hardware MainBoard CPLDFirmware: "s01770-6"
*s SystemUnit Hardware MainBoard Temperature: "33.5"
*s SystemUnit State Status: OK
*s SystemUnit State StatusCode: OK
*s SystemUnit State StatusText: ""
*s SystemUnit State H320Number: "21090"
*s SystemUnit State MaxNumberOfCalls: 3
*s SystemUnit State NumberOfActiveCalls: 1
*s SystemUnit State NumberOfSuspendedCalls: 0
*s SystemUnit State NumberOfInProgressCalls: 0
** end
```

xStatus SystemUnit ProductType

Shows the product type.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit ProductType
*s SystemUnit ProductType: "Cisco TelePresence ISDN link"
** end
```

xStatus SystemUnit Uptime

Shows the number of seconds since the last restart of the system.

Value space of the result returned:

<Integer>

Example:

```
xStatus SystemUnit Uptime
*s SystemUnit Uptime: 2158
** end
```

xStatus SystemUnit Software Version

Shows the software version installed on the system.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software Version
*s SystemUnit Software Version: "IL1.0.0"
** end
```

xStatus SystemUnit Software Name

Shows the name of the software that is installed on the system.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software Name
*s SystemUnit Software Name: "s51500"
** end
```

xStatus SystemUnit Software ReleaseDate

Shows the release date of the software installed on the system.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software ReleaseDate
*s SystemUnit Software ReleaseDate: "2012-05-30"
** end
```

xStatus SystemUnit Software BootStage1

Shows the version of boot stage 1 loader.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software BootStage1
*s SystemUnit Software BootStage1: "U-boot Stage1 2010.06-4"
** end
```

xStatus SystemUnit Software BootStage2

Shows the version of the boot stage 2 loader.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software BootStage2
*s SystemUnit Software BootStage2: "U-boot Stage2 2010.06-8"
** end
```

xStatus SystemUnit Software MaxVideoCalls

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

Value space of the result returned:

<Integer>

Example:

```
xStatus SystemUnit Software MaxVideoCalls
*s SystemUnit Software MaxVideoCalls: 3
** end
```

xStatus SystemUnit Software MaxAudioCalls

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

Value space of the result returned:

<Integer>

Example:

```
xStatus SystemUnit Software MaxAudioCalls
*s SystemUnit Software MaxAudioCalls: 3
** end
```

xStatus SystemUnit Hardware Module SerialNumber

Shows the serial number of the system.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Hardware Module SerialNumber
*s SystemUnit Hardware Module SerialNumber: "FTT1234567V"
** end
```

xStatus SystemUnit Hardware Module Identifier

Shows the revision of the system.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Hardware Module Identifier
*s SystemUnit Hardware Module Identifier: "01"
** end
```

xStatus SystemUnit Hardware MainBoard SerialNumber

Shows the serial number of the main board in the system.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Hardware MainBoard SerialNumber
*s SystemUnit Hardware MainBoard SerialNumber: "F0999989"
** end
```

xStatus SystemUnit Hardware MainBoard Identifier

Shows the revision of the main board in the system.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Hardware MainBoard Identifier
*s SystemUnit Hardware MainBoard Identifier: "102030-1 [00]"
** end
```


xStatus SystemUnit Hardware MainBoard CPLDFirmware

Shows the version of installed CPLD firmware.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Hardware MainBoard CPLDFirmware
*s SystemUnit Hardware MainBoard CPLDFirmware: "s01770-6"
** end
```

xStatus SystemUnit Hardware MainBoard Temperature

Shows the temperature (in degrees Celsius) of the system's main board.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Hardware MainBoard Temperature
*s SystemUnit Hardware MainBoard Temperature: "33.5"
** end
```

xStatus SystemUnit State Status

Shows the system state, i.e. whether there are any errors or warnings issued.

Initializing: The system is initializing and not yet ready for use.

OK: The system is ready for use.

OKWithWarning: The system can be used for making calls, but there are warnings issued that might influence its behavior.

Error: The system is in an erroneous state and cannot be used for making calls.

Value space of the result returned:

<Initializing/OK/OKWithWarning/Error>

Example:

```
xStatus SystemUnit State Status
*s SystemUnit State Status: Error
** end
```

xStatus SystemUnit State StatusCode

Shows the type of error or warning that is issued for the system (if any).

OK: The system is ready for use.

EthLinkDown: There is no Ethernet connection, and thus the ISDN Link cannot be used.

BRIWarning: A warning is issued for the BRI interface.

BRIError: There is an error on the BRI interface.

PRIError: There is an error on the PRI interface.

ExtNetError: There is an error on the external network interface (NET).

Initializing: The system is initializing and not yet ready for use.

Value space of the result returned:

<OK/EthLinkDown/BRIWarning/BRIError/PRIError/ExtNetError/Initializing>

Example:

```
xStatus SystemUnit State StatusCode
*s SystemUnit State StatusCode: BRIError
** end
```

xStatus SystemUnit State StatusText

Textual description of the system state, e.g. type of alarm if in an erroneous state.

Value space of the result returned:

<String>

Example:

```
xstatus SystemUnit State StatusText
*s SystemUnit State StatusText: "BRI1: [L1 Alarm] BRI2: [L1 Alarm] BRI3: [L1 Alarm] BRI4: [L1 Alarm]"
** end
```

xStatus SystemUnit State H320Number

Shows the number that can be used to dial in on the ISDN BRI/PRI interface in order to reach this ISDN Link and subsequently the endpoint associated with the ISDN Link. This requires that numbers have been assigned to the BRI or PRI interfaces respectively.

Not applicable when using the external network (NET) interface.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit State H320Number
*s SystemUnit State H320Number: "5678"
** end
```

xStatus SystemUnit State MaxNumberOfCalls

Shows the maximum number of simultaneous calls.

Value space of the result returned:

<1..3>

Example:

```
xStatus SystemUnit State MaxNumberOfCalls
*s SystemUnit State MaxNumberOfCalls: 3
** end
```

xStatus SystemUnit State NumberOfActiveCalls

Shows the number of active calls.

Value space of the result returned:

<0..3>

Example:

```
xStatus SystemUnit State NumberOfActiveCalls
*s SystemUnit State NumberOfActiveCalls: 1
** end
```

xStatus SystemUnit State NumberOfSuspendedCalls

Shows the number of suspended calls.

Value space of the result returned:

<0..3>

Example:

```
xStatus SystemUnit State NumberOfSuspendedCalls
*s SystemUnit State NumberOfSuspendedCalls: 0
** end
```

xStatus SystemUnit State NumberOfInProgressCalls

Shows the number of calls in progress.

Value space of the result returned:

<0..3>

Example:

```
xStatus SystemUnit State NumberOfInProgressCalls
*s SystemUnit State NumberOfInProgressCalls: 0
** end
```



On our web site you will find an overview of the worldwide Cisco contacts.

Go to: <http://www.cisco.com/web/siteassets/contacts>

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

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

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

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

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

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

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

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

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