
Nortel Communication Server 1000

Nortel Communication Server 1000 Release 4.0

CS 1000 Release 4.0 Troubleshooting Guide for Distributors

Expert Guide

Document Release: Standard 2.00

Date: February 2006

Copyright © 2006 Nortel Networks. All Rights Reserved.

Produced in Canada

The information in this document is subject to change without notice. The statements, configurations, technical data, and recommendations in this document are believed to be accurate and reliable, but are presented without express or implied warranty. Users must take full responsibility for their applications of any products specified in this document. The information in this document is proprietary to Nortel Networks.

Nortel, Nortel (Logo), the Globemark, SL-1, Meridian 1, and Succession are trademarks of Nortel Networks.

Revision history

February 2006

Standard 2.00. This document is up-issued for Communication Server 1000 Release 4.0.

December 2003

Issue 1.2. This document is up-issued for Succession release 3.0.

December 2002

Issue 1.1. This document is up-issued for the next release of the product.

October 2002

Issue 1.0. This is the first issue of this document.

Contents

About this document	7
Subject	7
Related information	7
Abbreviations	9
Introduction	15
General guidelines	15
System requirement	16
General troubleshooting	16
Problem scenarios	16
Signaling Server base software	17
Signaling Server boot-up	17
Signaling Server base commands	22
VxWorks shell commands	45
Command reference for Signaling Server base	54
Signaling Server tasks	136
Signaling Server Quality of Service (QoS)	138
Voice Gateway troubleshooting	141
Enhanced trace tools	141
IP Peer Networking and Gatekeeper	147
Software components	147

Network Routing Server	148
SIP Redirect Server	148
Gatekeeper	155
Gatekeeper software	163
Virtual trunk on the Call Server	168
Virtual Trunk on the Gateway	178
Signaling Server installation	189
Directory structure on the software CDROM	189
Install Tool initialization	191
Signaling Server software installation	196
VGMC loadware copy	197
Internet Telephone firmware copy	197
Basic configuration	197
Changing the Signaling Server configuration	200
Troubleshooting Problem Conditions	201
Element Manager	211
EM components	212
Data networking	231
Call Server and Media Gateway (SIPE) diagnostics	231
IP Line 4.0 Troubleshooting	235

About this document

This document is a global document. Contact your system supplier or your Nortel representative to verify that the hardware and software described are supported in your area.

Subject

This document describes troubleshooting techniques for Communication Server 1000 Release 4.0. The document is intended to provide additional information for distributors and technicians already familiar with the content of the Release 4.0 NTPs.

Note on legacy products and releases

This document contains information about systems, components, and features that are compatible with Nortel Communication Server 1000 Release 4.0 software. For more information on legacy products and releases, click the **Technical Documentation** link under **Support & Training** on the Nortel home page:

www.nortel.com

Related information

Online

To access Nortel documentation online, click the **Technical Documentation** link under **Support & Training** on the Nortel home page:

www.nortel.com

CD-ROM

To obtain Nortel documentation on CD-ROM, contact your Nortel customer representative.

Abbreviations

ACF	- Admission Confirmation
ARP	- Address Resolution Protocol
ARQ	- Admission Request
BCM	- Business Communication Manager
BIOS	- Basic Input/Output System. A set of permanently stored program routines.
BOOTP	- BOOTstrap Protocol
CDP	- Coordinated Dialing Plan
CGI	- Common Gateway Interface
CLAN	- Customer LAN. Regular data LAN for customer.
CLI	- Command Line Interface
CODEC	- COder DECoder. A DSP function that converts the PCM stream to packets and back.
CPU	- Central Processing Unit
CS	- Call Server
CSU	- Centralized (automatic) Software Upgrade
DCF	- Disengage Confirmation

DCH	- D Channel
DHCP	- Dynamic Host Configuration Protocol
DMI	- Desktop Management Interface
DN	- Directory Number
DRJ	- Disengage Reject
DRQ	- Disengage Request
DSCP	- Differentiated Services Code Point(DiffServe Code Point)
DSP	- Digital Signal Processor
DTMF	- Digitone Multifrequency/Dual Tone Multiple Frequency
EC	- Echo Canceller
ELAN	- Embedded LAN (10 BaseT LAN connection for internal signaling)
EM	- Element Management
EXUT	- Enhanced Extended Universal Trunk
FTP	- File Transfer Protocol
F/W	- Firmware
GCF	- Gatekeeper Confirmation
GK	- Gatekeeper
GRJ	- Gatekeeper Reject
GRQ	- Gatekeeper Request
GW	- Gateway
ICMP	- Internet Control Message Protocol
IP	- Internet Protocol
IPE	- Intelligent Peripheral Equipment
ISDN	- Integrated Services Data Networking
ITG	- Meridian Integrated IP Telephony Gateway

LAN	- Local Area Network
LCN	- Location Confirmation
LRJ	- Location Reject
LRQ	- Location Request
MAT	- Meridian Administration Terminal - Windows application for configuring the Meridian-1 PBX, ITGL and ITGT applications.
MCDN	- Meridian Customer Defined Network
MG	- Media Gateway
NPA	- Numbering Plan Area (area code)
NPI	- Numbering Plan Identification
NPM	- Network Protocol Module
OAM	- Operations, Administration, Maintenance - OAM shell
OM	- Operational Measurement
ODBC	Open Database Connectivity
OTM	- Optivity Telephony Manager - replacement for MAT
OS	- Operating System (VxWorks)
PC Card	- Current name used for PCMCIA style cards
PCM	- Pulse Code Modulation
PCMCIA	- Personal Computer Memory Card International Association. This organization has defined a credit card sized plug in board for use in PCs.
PDT	- Problem Determination Tools - PDT shell
PPP	- Point to Point Protocol
PSTN	- Public Switched Telephony Network
QoS	- Quality of Service
RAS	- Registration, Admission and Status

- RCF** - **Registration Confirmation**
- RLR** - **Receive Loudness Rating**
- RPC** - **Remote Procedure Call**
- RRJ** - **Registration Reject**
- RRQ** - **Registration Request**
- RTC** - **Real Time Clock**
- RTP** - **Real Time Protocol. A layer above UDP for synchroni-
zation of voice packets.**
- RTCP** - **Real Time Control Protocol. Used to exchange end point
statistics about a voice packet stream.**
- RUDP** - **Reliable User Datagram Protocol.**
- SA** - **StrongArm processor on Succession Media Card**
- SM** - **Subnet Mask**
- SMC** - **Succession Media Card (8 or 32 port, single slot card for
IP-TDM transcoding). In earlier releases, this is known
as the ITG-SA card**
- SS** - **Signaling Server**
- SLR** - **Send Loudness Rating**
- S/W** - **Software**
- SNMP** - **Simple Network Management Protocol. Used to raise
alarms and communicate OA&M between MAT/OTM
and the ITG card.**
- SNTP** - **Simple Network Time Protocol. Used to synchronize
time between cards.**
- STMR** - **SideTone Masking Rating**
- TCB** - **Task Control Block**
- TCID** - **Telephony Channel Identifier**
- TCP** - **Transmission Control Protocol**

TDM	- Time Division Multiplexing
TFTP	- Trivial File Transfer Protocol
TLAN	- Telephony LAN (10/100 BaseT LAN connection for ITG's RTP packets)
TN	- Terminal Number
ToS	- Type of Service
TON	- Type Of Number
TPS	- Terminal Proxy Server
UCF	- Unregister Confirmation
UDP	- Uniform Dialing Plan
UDP	- User Datagram Protocol
UIPC	- Universal ISDN Protocol Converter
UNIStim	- Unified Networks IP Stimulus protocol. Signalling protocol used between the ITGL and etherset.
URL	- Uniform Resource Locator
URQ	- Unregister Request
VCM	- Virtual Connection Manager
VGMC	- Voice Gateway Media Card
VGW	- Voice Gateway
VoIP	- Voice over IP
VTRK	- Virtual Trunk
WAN	- Wide Area Network
XUT	- Universal Trunk Card (Analog Trunk)
XML	- eXtensible Markup Language

Introduction

This document contains troubleshooting tools and techniques for CS 1000 Release 4.0.

This information is intended for distributors and on-site technicians who are already familiar with the information in the CS 1000 Release 4.0 NTPs (particularly *IP Peer Networking: Installation and Configuration* (553-3001-213)), and have hands-on experience in configuring IP Peer Networking. The document provides additional information to aid in problem troubleshooting.

This document is organized into sections, one for each component of the CS 1000 Release 4.0 system. Each section provides detailed troubleshooting information.



CAUTION — Service Interruption

Some of the commands described in this document may cause the system to become inoperable. If incorrect information is entered, a manual reboot may be required to recover. Alternatively, the system may reboot on its own. Use caution when employing these commands on a live system.

General guidelines

Each section provides troubleshooting commands for a particular component of the system.

The description of each command includes the command syntax and an explanation of the command function. In many cases, examples are also provided.

The commands are case sensitive. Parameters for the commands are provided within <angle brackets>.

Execute the commands from the specified shell only.

System requirement

For CS 1000 Release 4.0, configure the Signaling Server with at least 512 Mbytes RAM.

General troubleshooting

Network level troubleshooting

The *IP Line Release 4.0 Troubleshooting Guide for Distributors* provides information about network problem conditions related to voice quality, telephone resets and speech paths. Refer to the document for further details.

Problem scenarios

Voice quality problems

Voice quality problems are generally related to Media Cards and DSP functionality. Common symptoms include garbled voice and choppy speech conditions. See *IP Line Release 4.0 Troubleshooting Guide for Distributors* for information about troubleshooting these types of problems.

Signaling Server base software

The Signaling Server (SS) base software provides the following functionalities:

- login service,
- event logging,
- patching functionality,
- rlogin service,
- telnet service, and the
- object module loader.

Signaling Server boot-up

This section provides details about Signalling Server boot-up.

Initial BIOS screens

A full serial cable is required to see the BIOS boot screens. The hardware flow control is enabled during BIOS boot. The maintenance terminal can emulate VT100 and ANSI. The default terminal speed is 19200 bits per second.

The following is a sample boot screen:

```
AMIBIOS (C)2001 American Megatrends Inc.  
Copyright 1996-2001 Intel Corporation
```

```
TR440BXA.86B.0042.P15.0107200951
```

**Intel(R) Pentium(R)III processor, 700MHz
256MB OK**

Hit <F2> if you want to run SETUP

The BIOS version must be at least P15 (TR440BXA.86B.0042.**P15**.0107200951). Some terminals do not pass function keys (for example, F2 for SETUP). In this case, connect a PC keyboard directly to the Signalling Server.

Boot sequence

- 1** The BIOS is configured with the order of boot devices and to load the boot track.
- 2** The boot track (boot loader or boot ROM) is read from floppy, CDROM, or hard disk.

Note: Boot parameters are saved in the **nvr.am.sys** file, which can exist on any boot device (floppy, CDROM, or hard disk). Any changes to the boot parameters are saved in the **nvr.am.sys** file (unless the device is read only).

- 3** The boot ROM loads and starts the main load from the floppy, CDROM, hard disk, or the network.

As shown in the following example, the system prompts you for the device from which to read the boot parameters. If you enter **[C]** in response, the boot parameters are read from the CDROM. If you enter **[H]** in response, the boot parameters are read from the hard disk. If a selection is not made, the menu times out and the default device, **[H]**, is selected.

This sequence occurs only if there is a bootable CDROM or a boot floppy disk as well as a bootable hard disk in the system.

Read the boot parameters from:

```
[C]DROM
[H]ard Disk
3 [H]
```

Reading boot parameters from /p/nvram.sys

Press any key to stop auto-boot...

0
auto-booting...

If you unintentionally stop the auto boot menu by typing a character, use the @ command to continue the boot operation.

Boot order

The following is the boot order of the boot devices. Software boots automatically reset to this order.

- 1 ATAPI CDROM
- 2 IDE-HDD
- 3 floppy

General boot parameters

The boot parameters required during boot-up are:

- boot device (interface)
- the main OS file to load (main OS module)
- ELAN network parameters (hostname, IP, Subnet Mask, Gateway)
- network (FTP) server parameters (hostname, IP)
- boot options flag (see Figure 1: “Boot options flag bits” on [page 19](#))
- startup script (not usually used)

Figure 1: “Boot options flag bits” on [page 19](#) illustrates the bits that constitute the boot flag.

Figure 1
Boot options flag bits

M3	M2	M1	M0	0	0	0	0	0	0	0	0	L3	L2	L1	L0
----	----	----	----	---	---	---	---	---	---	---	---	----	----	----	----

The values and associated meanings of the boot flag bits are:

- M3 = 1—do not load or run application programs
- M3 = 0—load application programs
- M2 = 1—not used
- M1 = 1—Signaling Server is leader
- M1 = 0—Signaling Server is follower
- M0 = 1—install
- L1 = 1—load all symbols (including static, local and private symbols)
- L1 = 0—do not load static, local, and private symbols

Leader boot parameters

To identify a leader Signaling Server, the values of the boot parameters are:

- boot device: ata drive 0, controller 0
- file name of **mainos.sys**, on the hard disk
- ELAN IP and subnet mask (SM)
- ELAN gateway
- flags—set to 0x2000
- target name (my host name)
- other—set to fei (primary network interface (ELAN))

The following example illustrates the boot parameters required to identify a Signaling Server as a leader.

```
boot device      : ata=0,0
unit number     : 0
processor number : 0
file name       : /p/mainos.sys
inet on ethernet (e) : 47.11.216.XX:fffff00
gateway inet (g)  : 47.11.216.1
flags (f)        : 0x2000
target name (tn) : my_ss
other (o)        : fei
```

Follower boot parameters

The values of the boot parameters that identify a Signaling Server as a follower are:

- boot device: ata drive 0, controller 0
- file name of **mainos.sys**, on the hard disk
- network parameters are obtained using BOOTP
- flags—set to 0x0
- target name (my host name)
- other—no parameter value

The following example illustrates the boot parameters required to identify a Signaling Server as a follower.

```
boot device      : ata=0,0
unit number     : 0
processor number : 0
file name       : /cd0/mainos.sys
flags (f)       : 0x0
target name (tn) : my_follower_ss
```

CDROM (install) boot parameters

The boot parameter values for a CDROM are:

- boot device: ata drive 1, controller 0
- file name (of **mainos.sys**, on CDROM)
- no network parameters
- flags—set to 0x100a (installation)
- other—no parameter value

The following example illustrates the boot parameters for a CDROM,

```
boot device      : ata=1,0
unit number     : 0
processor number : 0
```

file name : /cd0/mainos.sys
flags (f) : 0x100a

Signaling Server base commands

This section describes the commands that you can use to retrieve information from the Signaling Server.

The **telnet**, **rlogin**, and **cslogin** commands are available in the OAM, PDT and VxWorks shells. Using the **cslogin** command, you can **rlogin** from the Signaling Server to the Call Server (CS) and access the overlay supervisor without knowing the CS PDT password.

OAM shell commands

The commands available in the OAM shell are provided by different applications. Type **help** to display the commands of the various applications. The OAM shell is available upon successful login to the Signaling Server. The OAM shell environment is identified by the **oam>** prompt.

The following system administration commands are available in the OAM shell. Section “OAM shell commands—examples” on [page 30](#) provides examples of output from these commands.

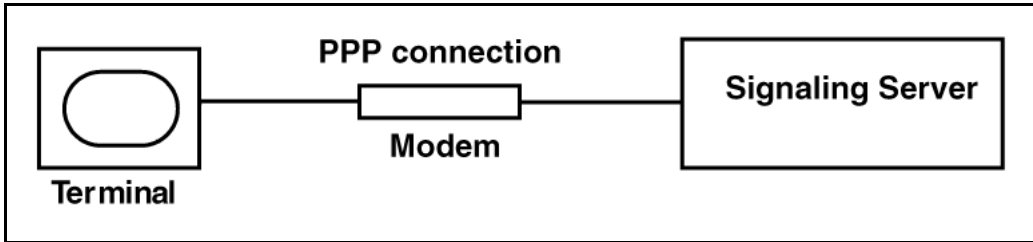
- **telnet <IP address or hostname>**
Telnet to the server specified by IP address or hostname.
- **rlogin<IP address or hostname>**
Remote log in to the server specified by IP address or hostname.
- **cslogin**
Log in to the Call Server overlays.
- **routeShow**
Display host and network routing tables.
- **routeAdd <destination><gateway>**
Add a new route with destination and gateway to the routing tables.

- **arpShow**
Display the system ARP table.
- **arpFlush**
Flush all entries in the system ARP table.
- **swVersionShow**
Display the software version of the Signaling Server.
- **date <> or <day month date hh:mm:ss year>**
With no parameters specified, display the current date. With parameters, set the time and date to the given parameters.
- **stty <port speed>**
Set the maintenance port speed. The values for the maintenance port speed are 9600, 19200, 38400 and 115200 baud.
- **ppp -l <local IP address> -r <remote IP address> -o <options file>**
Set up a point-to-point connection from a terminal to the Signaling Server. The details of options available for debugging purposes are available in VxWorks documentation.
- **who**
Display the number and the identity of users connected to the Signaling Server.

Setting up a point-to-point connection to the Signaling Server

Figure 2: “Point to point connection” on [page 24](#) illustrates a simple configuration for setting up a PPP connection to the Signaling Server.

Figure 2
Point to point connection



The Signaling Server has two maintenance ports, one on the front face plate and the other on the back panel. The modem can be connected to either of the maintenance ports. The front maintenance port will not display system messages.

To set up the point-to-point (PPP) connection from the terminal, type the command **ppp** in a terminal window. The Signaling Server will use IP addresses assigned by default for this connection.

Table 1: “IP Addresses for PPP sessions” on [page 24](#) provides the default IP addresses that are assigned by the Signaling Server for the PPP session based on the maintenance ports used.

Table 1
IP Addresses for PPP sessions

Maintenance port on SS	Local IP address for SS	Remote IP address for terminal
Back panel	137.135.3.1	137.135.3.2
Front panel	137.135.5.1	137.135.5.1

To use specific IP addresses for setting up the PPP connection, enter the following command in the terminal window:

- **ppp -l <SSIPaddr> -r <TermIPaddr>**

where **SSIPaddr** is the IP address assigned to the Signaling Server and **TermIPaddr** is the IP address assigned to the terminal.

Services switch over

This section lists the commands available for services switch over. These commands are available in the OAM shell.

- **soHelpMenu**

Lists the commands available for services switch over.

- **soCmdStatusShow**

Displays the status of the service switch over commands.

Graceful disable commands

This section describes the graceful disable commands for the IP Line, Virtual Trunk and Gatekeeper applications. These commands are available in the OAM shell and do not interrupt established calls. When Graceful commands are executed, the system determines if there are available resources to reregister such resources as virtual trunks (VTRK) and IP telephones, and instructs the VTRKs and telephones to reregister only when they become idle.

- **disServices**

Gracefully switch the registered resources from the VGMC card or Signaling Server to the other VGMC cards or Signaling Servers located in the same node.

- **disVTRK**

Gracefully switch the registered virtual trunks from the Signaling Server to another Signaling Server located in the same node.

- **disTPS**

Gracefully switch the registered line TPS from the VGMC cards or Signaling Server to the other VGMC cards or Signaling Servers located in the same node.

- **disGK**

Switch the local GK out of service and the alternative GK into service.

Force disable commands

This section describes the force disable commands for the IP Line, Virtual Trunk and Gatekeeper applications. These commands are available from the OAM shell.

The force commands instruct the system to unregister the virtual trunks and telephones regardless of the availability of another VGMC or Signaling Server to which these resources can reregister. Resources unable to reregister are stranded (stranded telephones constantly reboot and stranded virtual trunks are disabled, making trunk calls over them impossible). The force commands also tear down active calls (trunk calls using VTRK resources are dropped and telephones on a call are reset).

- **forcedisServices**

Force all registered resources on the VGMC or Signaling Server to unregister and force the Gatekeeper to go out of service

- **forcedisVTRK**

Force all registered virtual trunks to unregister from the local server.

- **forcedisTPS**

Force all telephones registered to the local line TPS to unregister

- **forcedisGK**

Force the local Gatekeeper to go out of service

Enable commands

This section describes the enable commands for the IP Line, Virtual Trunk and Gatekeeper applications. These commands are available in the OAM shell.

- **enlServices**

Force all the VGMC cards or Signaling Server to accept registrations of resources.

- **enlVTRK**

Force the Signaling Server to accept virtual trunk registrations. The virtual trunks are only registered to the master Signaling Server of the node. If the Signaling Server being enabled becomes the new

master, the configured virtual trunks will re-register to that Signaling Server.

- **enITPS**

Enable the TPS application and the TPS registration process on the system line TPS. Applies to both the VGMC and the Signaling Server.

- **enIGK**

Force the local Gatekeeper to go in service. The local Gatekeeper becomes active under one of the following conditions:

- It is configured as the prime Gatekeeper.
- It is configured as the alternative Gatekeeper and the prime Gatekeeper is out of service.
- It is configured as the fail safe Gatekeeper and both the prime Gatekeeper and the alternative Gatekeeper are out of service.

The local Gatekeeper goes into standby under one of the following conditions:

- It is configured as the alternative Gatekeeper and the prime Gatekeeper goes back into service (becomes active).
- It is configured as fail safe and the prime Gatekeeper or the alternative Gatekeeper goes back into service (becomes active)

Other commands

This section describes other useful commands available in the OAM shell.

- **loadBalance**

Cause the VGMC card or Signaling Server to attempt to balance the registration load of telephones between this card/server and the rest of the node components. If there is a Signaling Server in the node, no SMC or ITG-P will take over any telephones. This command

only applies to a group of servers of the same type (Signaling Servers, ITG-Ps, and SMCs, for example).

- **servicesStatusShow**

Display the status of all the services configured in the local platform.

Trace and diagnostic commands

H323 trace commands

This section describes the OAM commands available for tracing H323 messaging on active trunk calls.

- **H323CallTrace <trace_state>**

Activate or deactivate H.323 message tracing for all incoming and outgoing messages for all channels. Values for trace_state are ON and OFF.

- **H323CallTrace <MsgRecv> <MsgSend>**

Activate or deactivate H.323 message tracing for the incoming and/or outgoing messages for all channels where:

<MsgRecv> specifies activation or deactivation of tracing for H.323 messages . Values are ON or OFF;

<MsgSend> specifies activation or deactivation of tracing for H.323 messages. Values are ON or OFF.

- **H323CallTrace <channelNum> <MsgRecv> <MsgSend>**

Trace the incoming and/or outgoing messages for a specific channel where:

<channelNum> is the channel number. Values range from 0 - maximum channel number;

<MsgRecv> specifies activation or deactivation of tracing for H.323 messages sent to the specified channel. Values are ON or OFF.

<MsgSend> specifies activation or deactivation of tracing for H.323 messages sent from the specified channel. Values are ON or OFF.

- **H323CallTrace** <start_chNum> <end_chNum> <MsgRecv>
<MsgSend>

Activate or deactivate H.323 message tracing for a range of channels where:

<start_chNum> is the first channel number in range. Values range from 0 - maximum channel number;

<end_chNum> is the last channel number in the range. Values range from 0 - maximum channel number, but it must be greater than <start_chNum>;

<MsgRecv> specifies activation or deactivation of tracing for H323 messages sent to the specified range of channels. Values are ON or OFF;

<MsgSend> specifies activation or deactivation of tracing for H.323 messages sent from the specified range of channels. Values are ON or OFF.
- **H323Output** <output_destination> <on or off>

Direct the H323 trace output to the specified output destination.

Values for <output_destination> are:

1—TTY

2—RPTLOG
- **H323TraceShow**

Display the **H323CallTrace** and **H323Output** settings.

DCH diagnostic tool

The **DCHmenu** command is available from the OAM shell on the Signaling Server.

- **DCHmenu**

Display a menu of DCH diagnostic tools.

Maintenance terminal

The maintenance terminal port speed can be configured. The port speed is saved in the BIOS and survives reboots and power cycles. The **stty** command

sets the port speed on the maintenance ports on the front and back of the Signaling Server.

- **stty <speed>**

Set the console speed. The available speeds are 9600, 19200, 38400 and 11520 baud.

OAM shell commands—examples

This section provides examples of output from the OAM shell commands. The commands are listed in alphabetical order. See section “Command reference for Signaling Server base” on [page 54](#) for commands not listed in this section.

```
oam>date
MON OCT 28 18:00:48 2002
```

```
oam> date Tue OCT 29 10:00:00 2002
```

```
oam> date
TUE OCT 29 10:00:01 2002
```

```
oam> DCHmenu
Please select one of the DCHmenu options:
```

- 0 - Print menu (default)
- 1 - Print current DCH state
- 2 - Print current DCH configuration
- 3 - Print application error log
- 4 - Print link error log
- 5 - Print protocol error log
- 6 - Print message log
- 7 - Enable printing all messages processed by UIPC
- 8 - Enable error printing
- 9 - Enable info printing
- 10 - Enter manual message mode
- 11 - Print b channel control blocks
- 99 - Exit menu

Please enter your DCHmenu choice (0 to print the menu):

oam> disGK
oam> 03/12/03 03:32:47 LOG0006 tSOGK: disGK: Puts out of service the local gatekeeper and puts in service the alternative gatekeeper if available
03/12/03 03:32:47 LOG0006 tSOGK: The disGK command has failed. Reason: Unreachable Alternative GK. Available command: forcedisGK

oam> disServices
oam> 03/12/03 03:36:57 LOG0006 VTRK: The disVTRK command has failed. Reason: There are not enough virtual trunk resources in the node to handle reregistration of resources. Available command: forcedisVTRK
03/12/03 03:36:58 LOG0006 tSO: disGK: Puts out of service the local gatekeeper and puts in service the alternative gatekeeper if available
03/12/03 03:36:58 LOG0006 tSO: The disGK command has failed. Reason: Unreachable Alternative GK. Available command: forcedisGK
03/12/03 03:36:58 LOG0006 TPS: Virtual trunk application is active. First disable virtual trunks and then reissue disTPS command

oam> disTPS
oam> 03/12/03 03:32:06 LOG0006 TPS: Virtual trunk application is active. First disable virtual trunks and then reissue disTPS command

oam> disVTRK
oam> 03/12/03 03:27:13 LOG0006 VTRK: The disVTRK command has failed. Reason: There are not enough virtual trunk resources in the node to handle reregistration of resources. Available command: forcedisVTRK

oam> enlGK
oam> 03/12/03 03:25:29 LOG0006 tSOGK: enlGK: GK is already in service

oam> enlServices
oam> 05/12/03 10:02:11 LOG0006 shell: Causes the virtual trunk application to be enabled and to accept virtual trunk registrations
05/12/03 10:02:11 LOG0006 shell: vtrkStateHandler: VtrkSOEnable: No state change (state = VtrkStandby)
05/12/03 10:02:11 LOG0006 shell: enlGK: Causes the local gatekeeper to be put in service

05/12/03 10:02:11 LOG0006 GKNPM: In Service: Switching to GK_STANDBY
05/12/03 10:02:11 LOG0006 shell: GK in service
05/12/03 10:02:11 LOG0006 TPS: Service enabled
05/12/03 10:02:11 LOG0006 SET: Service enabled

oam> enLTPS
oam> 03/12/03 03:38:16 LOG0006 shell: LTPS is enabled, enLTPS was ignored

oam> enLVTRK
oam> 03/12/03 03:44:44 LOG0006 shell: Causes the virtual trunk application to be enabled and to accept virtual trunk registrations
03/12/03 03:44:44 LOG0006 shell: vtrkStateHandler: VtrkSOEnable: No state change (state = VtrkStandby)
03/12/03 03:44:46 LOG0006 GKNPM: gkNpmHandleRRQRequest: GK_OUT_OF_SERVICE
03/12/03 03:44:46 LOG0005 GKNPM: RAS FAILURE: RAS_TYPE : RRQ,Reason=discoveryRequired,SrcIP=47.11.217.205:1719
03/12/03 03:44:59 LOG0003 tNetTask: [ARP] duplicate IP address 2f0bf98b sent from ethernet address 00:60:38:bd:06:31
03/12/03 03:44:59 LOG0006 CSV: CSV enable
03/12/03 03:44:59 LOG0006 CSV: Node 1999 registering for terminal connections on 47.11.249.139:4100
03/12/03 03:44:59 LOG0006 TPS: No security checking for this card
03/12/03 03:44:59 LOG0004 UMS: No firmware for i2001 was found in /u/fw/
03/12/03 03:44:59 LOG0006 ELC: VTRK: This signal server is master
03/12/03 03:44:59 LOG0006 ELC: gkNpmCardEventHandler: unhandled event 0x4
03/12/03 03:44:59 LOG0006 VTRK: vtrkStateHandler: VtrkElecWon: State change from VtrkStandby to VtrkRegistration
03/12/03 03:44:59 LOG0006 VTRK: server SSRC is 2048
03/12/03 03:44:59 LOG0006 VTRK: leaderFlag is 16777216
03/12/03 03:44:59 LOG0006 VTRK: after htonl ssrc is 524288
03/12/03 03:44:59 LOG0006 VTRK: server <LeadeSS> node <1999> online announce
03/12/03 03:44:59 LOG0006 NPM: npmMasterUpdate: register with GK
03/12/03 03:44:59 LOG0005 ELC: Election won, master = 47.11.249.176
03/12/03 03:44:59 LOG0006 GKNPM: gkNpmHandleRRQRequest: GK_OUT_OF_SERVICE
03/12/03 03:44:59 LOG0005 GKNPM: RAS FAILURE: RAS_TYPE : RRQ,Reason=discoveryRequired,SrcIP=47.11.249.139:1719

03/12/03 03:44:59 LOG0003 NPM: cmEvRASReject: GK 47.11.249.176 Registration rejected
 03/12/03 03:44:59 LOG0003 NPM: cmEvRASReject: GK 47.11.215.243 Registration rejected
 03/12/03 03:44:59 LOG0003 VTRK: itgMsgSend to task 0xf800
 03/12/03 03:44:59 LOG0003 VTRK: vtrkCDSInfoHandler: send CDS Info to SIPNPM failed
 03/12/03 03:44:59 LOG0003 VTRK: itgMsgSend to task 0xf800
 03/12/03 03:44:59 LOG0003 VTRK: vtrkCDSInfoHandler: send CDS Info to SIPNPM failed
 03/12/03 03:44:59 LOG0006 VTRK: ServerStatus OK
 03/12/03 03:44:59 LOG0006 VTRK: vtrkStateHandler: VtrkCSRegOK: State change from VtrkRegistration to VtrkActive
 03/12/03 03:44:59 LOG0004 VTRK: DCH Established

oam> forcedisGK

oam> 03/12/03 03:40:28 LOG0006 tSOGK: forcedisGK: Forces the local gatekeeper to be put out of service
 03/12/03 03:40:28 LOG0004 tSOGK: gkNpmResultCheck: unhandle gkNpmMsgType 11 for OOS_NO_ALT_GK_CONFIG
 03/12/03 03:40:28 LOG0006 GKNPM: OOS Force: Switching to GK_OUT_OF_SERVICE
 03/12/03 03:40:29 LOG0006 GKNPM: gkNpmHandleRRQRequest: GK_OUT_OF_SERVICE

oam> forcedisServices

oam> 05/12/03 09:55:39 LOG0006 shell: Forces all registered virtual trunks to unregister from the local server
 05/12/03 09:55:39 LOG0006 shell: vtrkStateHandler: VtrkSOForceDisable: State change from VtrkActive to VtrkDeregistration
 05/12/03 09:55:39 LOG0006 shell: VTRK Unregister all
 05/12/03 09:55:39 LOG0006 shell: server <S_Campbell_SS> offline announce
 05/12/03 09:55:39 LOG0006 shell: forcedisGK: Forces the local gatekeeper to be put out of service
 05/12/03 09:55:39 LOG0003 GKNPM: OOS Force: Switching to GK_OUT_OF_SERVICE
 05/12/03 09:55:39 LOG0006 shell: GK out of service
 :

oam> forcedisTPS

oam> 03/12/03 03:43:44 LOG0006 TPS: Force disable TPS

03/12/03 03:43:44 LOG0006 SET: Service force disabled, Reset All Sets

oam> forcedisVTRK

oam> 03/12/03 03:41:21 LOG0006 shell: Forces all registered virtual trunks to unregister from the local server

03/12/03 03:41:21 LOG0006 shell: vtrkStateHandler: VtrkSOForceDisable: State change from VtrkActive to VtrkDeregistration

03/12/03 03:41:21 LOG0006 shell: VTRK Unregister all

03/12/03 03:41:21 LOG0006 shell: server <LeadeSS> offline announce

03/12/03 03:41:21 LOG0003 VTRK: ITG2106 DCH 15 Failure (91)

03/12/03 03:41:21 LOG0006 VTRK: vtrkStateHandler: VtrkLastTNUnreg: State change from VtrkDeregistration to VtrkStandby

03/12/03 03:41:21 LOG0006 CSV: CSV disabled

03/12/03 03:41:21 LOG0006 ELC: VTRK: This signal server is not a master

03/12/03 03:41:21 LOG0006 ELC: gkNpmCardEventHandler: unhandled event 0x4

03/12/03 03:41:21 LOG0004 VTRK: vtrkStateHandler: event 1 ignored ; state = VtrkStandby

03/12/03 03:41:21 LOG0006 NPM: npmMasterUpdate: registration timer killed

03/12/03 03:41:21 LOG0006 NPM: npmMasterUpdate: already unregistered

03/12/03 03:41:21 LOG0003 VTRK: itgMsgSend to task 0xf800

03/12/03 03:41:21 LOG0003 VTRK: vtrkCDSInfoHandler: send CDS Info to SIPNPM failed

03/12/03 03:41:21 LOG0003 VTRK: itgMsgSend to task 0xf800

03/12/03 03:41:21 LOG0003 VTRK: vtrkCDSInfoHandler: send CDS Info to SIPNPM failed

03/12/03 03:41:21 LOG0006 VTRK: ServerStatus OK

03/12/03 03:41:21 LOG0004 VTRK: vtrkStateHandler: event 6 ignored ; state = VtrkStandby

oam> H323CallTrace

H323CallTrace

Usage: A) H323CallTrace <on or off>

B) H323CallTrace <on or off (MsgRecv)> <on or off (MsgSend)>

C) H323CallTrace <channel #> <on or off (MsgRecv)> <on or off (MsgSend)>

D) H323CallTrace <start channel #> <end channel #> <on or off (MsgRecv)> <on or off (MsgSend)>

```
oam> H323TraceShow
TTY output: OFF
RPT output: ON
```

```
Channels H323MsgRecv (VTRK->NPM) H323MsgSend (NPM->VTRK)
=====
0 - 382      ON          ON
```

```
IPL> loadBalance
value = 0 = 0x0
IPL>
DEC 05 09:43:49 tShell: Info Line TPS is attempting to balance the registration
load of sets between this card and the rest of the node components
DEC 05 09:43:54 TPS: Info Average number of sets registered to a signaling
server should be: 0. Average number of sets registered to a VGMC card should
be: 3
DEC 05 09:43:54 SET: Info Load balance message received from TPS
```

```
oam> rlogin 47.11.255.29
```

```
login:
password:
```

```
oam> routeAdd 47.11.255.0 47.11.255.30
```

```
oam> routeShow
```

ROUTE NET TABLE

destination	gateway	flags	Refcnt	Use	Interface
0.0.0.0	47.11.249.1	3	0	88	fe1
47.11.249.0	47.11.249.111	101	0	0	fe1
47.11.254.0	47.11.255.29	101	0	0	fe1

ROUTE HOST TABLE

destination	gateway	flags	Refcnt	Use	Interface
-------------	---------	-------	--------	-----	-----------

```
47.11.255.0 47.11.255.30 7 0 0 fei0
127.0.0.1 127.0.0.1 5 1 1 lo0
```

oam> routeShow

ROUTE NET TABLE

destination	gateway	flags	Refcnt	Use	Interface
0.0.0.0	47.11.249.1	3	0	84	fei1
47.11.249.0	47.11.249.111	101	0	0	fei1
47.11.254.0	47.11.255.29	101	0	0	fei0

ROUTE HOST TABLE

destination	gateway	flags	Refcnt	Use	Interface
127.0.0.1	127.0.0.1	5	1	1	lo0

oam> soCmdStatusShow

VTRK Services Switch Over Command Status Show

Command: forcedisVTRK

Status: Successful

Reason: Forces all registered virtual trunks to unregister from the local server. All virtual trunk TNs will be reset within a couple of minutes

GK Services Switch Over Command Status Show

Command: forcedisGK

Status: Successful

Reason: Forces the local gatekeeper to be put out of service

LTPS Services Switch Over Command Status Show

Command: forcedisTPS

Status: Successful

Reason: Forces all registered line LTPS to unregister from the local server. All sets will be reset within a couple of minute

oam> soHelpMenu

Services Switch Over Help Menu

Graceful disable services

disServices: Causes the server to gracefully switch the registered resources to the other services in the same node

disTPS: Causes the line LTPS to gracefully switch the registered sets to the other cards located in the same node

disVTRK: Causes the virtual trunk to gracefully switch the registered virtual trunks to other SS located in the same node

disGK: Puts out of service the local gatekeeper and puts in service the alternative gatekeeper if available

Force disable services

forcedisServices: Forces the server to switch the registered resources to the other services in the same node

forcedisTPS: Forces all registered line LTPS to unregister from the local server

forcedisVTRK: Forces all registered virtual trunks to unregister from the local server

forcedisGK: Forces the local gatekeeper to be put out of service

Enable services

enlServices: Causes all the services to accept registration of resources

enlTPS: Causes line LTPS application to be enabled and to accept set registrations

enlVTRK: Causes the virtual trunk application to be enabled and to accept virtual trunk registrations

enlGK: Causes the local gatekeeper to be put in service

loadBalance: Causes the service to attempt to balance the registration load of sets between this service and the rest of the node services

servicesStatusShow: Shows the status of services (tps/iset/vtrk/gk)

soCmdStatusShow: Shows the service switch over commands status

oam>

oam> swVersionShow

sse-2.00.74 Wednesday October 16 2002 20:04:18 EDT

Loaded Modules:
share.obj sse-2.00.74
line.obj sse-2.00.74
trunk.obj sse-2.00.74
gk.obj sse-2.00.74
web.obj sse-2.00.74

oam> stty 9600
oam>

oam> telnet 47.11.255.29
Trying 47.11.255.29

login:
password:

oam> who
3 /tyCo/0 nobody
5 /tyCo/1 target
42 /pty/pty00.S target 47.11.181.81
->device id->device name->username->connected IP(if applicable)]

PDT shell commands

This section describes the commands available from the PDT shell. To enter the PDT shell, type the **CTRL+p-d-t** key sequence. The PDT shell environment is indicated by the prompt, **pdt>**.

Patch commands

The patching facility commands described in this section are available in the PDT shell. Type **help Patcher** at the **pdt>** prompt for help on using these commands. If you enter these commands in the vxshell, you will receive an error message.

The software patch files are downloaded from a workstation to the Signaling Server and stored in FLASH memory. A patch file is loaded into DRAM memory using the **pload** command.

Examples of output from the patch commands are given in section “Software patching” on [page 156](#).

- **pload <patch filename>**
Load the patch file specified by <patch filename> into memory.
- **poos <patch handle>**
Take the specified patch out of service.
- **pout <patch handle>**
Remove the specified patch from memory.
- **plis <patch handle>**
List details of the specified patch.
- **pstat < > or <patch handle>**
With no parameters, list the status of all active patches. With a parameter, list the status of the specified patch.
- **pnew <patch filename>**
Create a memory patch.

System commands

This section describes the system commands available in the PDT shell. Type **help system** at the **pdtd>** prompt to obtain a list of the available system commands. All OAM shell commands are also available in the PDT shell.

- **lkup <string>**
Look up the symbol identified by <string>.
- **devs**
Display the list of devices.
- **echo**
Echo the input.
- **hosts**
Display the hosts list.

- **memShow**
Display the memory usage.
- **ti <taskname> or <taskid>**
Display the task information for the specified task.
- **i <taskname> or <taskid>**
Display the task information for the specified task.
- **version**
Display the VxWorks version.
- **who**
Display all active rlogin user IDs and ports.
- **ifShow <> or <networkinterfacename>**
Display the specified network interfaces. With no parameters, display all network interfaces.
- **reboot <> or <-1>**
With no parameters, perform a warm restart. If the parameter has a value of -1, perform a cold restart.
- **ls <path <,long>>**
List the contents of the directory specified by <path>.
- **ll <path>**
List the contents of the directory specified by <path> in long format.
- **pwd**
Print the current default directory.
- **cd <path>**
Change the current default directory to the directory specified by <path>.
- **remove <path>**
Remove the file specified by <path>.

- **copy** <in> <, out>
Copy from the input file, <in>, to the output file, <out>.
- **rename** <old>, <new>
Rename or move the file specified by <old> to the name specified by <new>.
- **moduleShow**
Display the list of all loaded modules.
- **inetstatShow**
Display all of the active connections for the IP sockets.
- **tcpstatShow**
Display the statistics for the TCP protocol.
- **udpstatShow**
Display the statistics for the UDP protocol.
- **syslogShow**
Display the log level for all tasks.
- **syslogLevelSet** <tid> or <taskname><level>
For the task identified by <tid> or <taskname>, set the log level to the value specified by <level>. The level can be set to a number in the range of 0 to 7.

Report log file commands

Type **help rdtools** at the **pdt>** prompt to obtain a list of the commands available for displaying information from the report log file.

The report log files are stored in the directory **/u/rpt** with a filename format of **LOG000nn.RPT**, where nn are numbers. The higher the number represented by nn, the more recent the report log file.

- **rdopen** <> or <filename>
Open the report log file specified by <filename>. If no parameter is specified, open the most recent report log file.

- **rdgo <recordnumber>**

In the report log file, go to the absolute record specified by <recordnumber>.
- **rd <numberofsteps> <numberofrecords>**

In the report log file, move the number of steps specified by <numberofsteps>. Display the number of records specified by <numberofrecords>. The parameters <numberofsteps> and <numberofrecords> can be positive or negative numbers.
- **rds <numberofsteps> <numberofrecords>**

In the report log file, move the number of steps specified by <numberofsteps>. With symbolic dump, display the number of records specified by <numberofrecords>. The parameters <numberofsteps> and <numberofrecords> can be positive or negative numbers.
- **rdshow**

Display the general information about the report log file.
- **rdall**

Display all records in the report log file without symbolic dump.
- **rdtail <numberofrecords>**
- Without symbolic dump, display the most recent records in the report log file. The number of records displayed is specified by <numberofrecords>.
- **rdhead <numberofrecords>**

Without symbolic dump, display the oldest records in the report log file. The number of records displayed is specified by <numberofrecords>.
- **rdnext**

Open the next report log file in the list of generated log files.
- **rdprev**

Open the previous report log file in the list of generated log files.

Output from report log file commands

Sample output from each of the PDT shell report log file commands is shown below.

```
pdt> rdopen "/u/rpt/LOG00009.RPT"  
Reading /u/rpt/LOG00009.RPT
```

```
pdt> rdgo 230  
[0230] 28/10/02 15:20:00 LOG0006 NPM: npmControlTOSGet: H323 Control  
Layer3 TOS is: 0x28
```

```
pdt> rd 0, 5  
[0230] 28/10/02 15:20:00 LOG0006 NPM: npmControlTOSGet: H323 Control  
Layer3 TOS is: 0x28  
[0231] 28/10/02 15:20:00 LOG0004 NPM: npmH323Init: not master, abort  
[0232] 28/10/02 15:20:01 LOG0006 tRootTask: Task npmInit initialization suc-  
ceeded  
[0233] 28/10/02 15:20:01 LOG0006 NPM: tNpm task init successful  
[0234] 28/10/02 15:20:03 LOG0006 HTTP: SYSLOG initialised
```

```
pdt> rds 0, 5  
[0235] 28/10/02 15:20:03 LOG0006 HTTP: Memory file system initialised!  
[0236] 28/10/02 15:20:03 LOG0006 HTTP: Setup HTTP Aliasing  
[0237] 28/10/02 15:20:03 LOG0006 HTTP: Setup HTTP File System  
[0238] 28/10/02 15:20:03 LOG0006 HTTP: Setup Server Side Includes  
[0239] 28/10/02 15:20:03 LOG0006 HTTP: Load web server config file success-  
ful!
```

```
pdt> rds 10, 10  
[0250] 28/10/02 15:20:04 LOG0006 HTTP: Finish loading the Error Look up ta-  
ble index file: 7 edd  
[0251] 28/10/02 15:20:04 LOG0006 HTTP: Finish loading the Error Look up ta-  
ble index file: 8 err  
[0252] 28/10/02 15:20:04 LOG0006 HTTP: Finish loading the Error Look up ta-  
ble index file: 9 esn  
[0253] 28/10/02 15:20:04 LOG0006 HTTP: Finish loading the Error Look up ta-  
ble index file: 10 hwr  
[0254] 28/10/02 15:20:04 LOG0006 HTTP: Finish loading the Error Look up ta-
```

ble index file: 11 ini
[0255] 28/10/02 15:20:04 LOG0006 HTTP: Finish loading the Error Look up table index file: 12 itg
[0256] 28/10/02 15:20:04 LOG0006 HTTP: Finish loading the Error Look up table index file: 13 npr
[0257] 28/10/02 15:20:04 LOG0006 HTTP: Finish loading the Error Look up table index file: 14 ovl
[0258] 28/10/02 15:20:04 LOG0006 HTTP: Finish loading the Error Look up table index file: 15 pri
[0259] 28/10/02 15:20:04 LOG0006 HTTP: Finish loading the Error Look up table index file: 16 rpt

pd> rdtail 5
[269] 28/10/02 15:20:04 LOG0006 HTTP: Task httpd initialization succeeded
[268] 28/10/02 15:20:04 LOG0006 HTTP: Finish loading the Error Look up table index file: 25 tfc
[267] 28/10/02 15:20:04 LOG0006 HTTP: Finish loading the Error Look up table index file: 24 mph
[266] 28/10/02 15:20:04 LOG0006 HTTP: Finish loading the Error Look up table index file: 23 misp
[265] 28/10/02 15:20:04 LOG0006 HTTP: Finish loading the Error Look up table index file: 22 msdl

pd> rdhead 5
[0000] 28/10/02 14:50:21 LOG0006 tRootTask: alarmInit initialization succeeded
[0001] 28/10/02 14:50:21 LOG0006 tRootTask: ITG5000 Card initialized, all alarms cleared. (202)
[0002] 28/10/02 14:50:21 LOG0006 tRootTask: shareAnnounce initialization succeeded
[0003] 28/10/02 14:50:21 LOG0006 tRootTask: ELAN IP = 47.11.255.29
[0004] 28/10/02 14:50:21 LOG0006 tRootTask: itgCardInit initialization succeeded

pd> rdnext
Reading /u/rpt/LOG00002.RPT

pd> rdprev
Reading /u/rpt/LOG00001.RPT

VxWorks shell commands

VxWorks shell access

You can invoke the VxWorks shell from the PDT shell by typing the command **vxshell** at the **pd>** prompt.

You can also invoke the Signaling Server base commands in the vxshell environment by typing the commands at the vxshell prompt (->). See section “Signaling Server base commands” on [page 22](#) for more information about these commands.

Task logging commands

Use the commands described in this section to display and set the logging levels associated with tasks.

- **syslogShow** (pg 127)

List each task and the corresponding level at which logging is set. Table 2: “System Logging Levels” on [page 45](#) provides a description of the different logging levels and the corresponding logging level value.

Table 2
System Logging Levels

Level	Value
Emergency, system is unusable	0
Critical, critical conditions	1
Warning, warning conditions	2
Info, informational	3
Alert, action must be taken immediately	4
Error, error conditions	5

Table 2
System Logging Levels

Level	Value
Notice, normal but significant condition	6
Debug, debug level messages	7

- **syslogLevelSet** <task>, <level> (pg 127)

Control the printing of detailed information from a task. When a particular level of logging is set for a task, the information for the level selected and all the levels below the selected level are printed. To print information for multiple tasks, you must enter the command multiple times. The parameter <task> specifies the module for this entry of the command. The name of the task as printed by **syslogShow** or the **i** command is used.

File system and configuration commands

Use the commands described in this section to display information about the file system and system configuration.

- **ifTabShow** (pg 86)

Print the contents of the file system configuration table.

- **dumptab** (pg 60)

Display the contents of the BOOTP server database.

- **sysConfigFileShow** (pg 123)

Displays the system configuration file.

The active configuration files **CONFIG.INI** and **BOOTP.TAB** are present in **/u/config** directory. These files are created when the Signaling Server software is installed.

The backup files are **CONFIG.BAK**, **BOOTP.BAK**. The system automatically creates the backup files. If there is a file corruption in the active file, the system replaces it with the backup file.

The configuration files can be printed to the console using the command:

```
->copy <configfilename>
```

Module and release information commands

Use the commands described in this section to display information about the modules and the release.

- **sysRlsModuleShow** (pg 129)

Display the module version.

- **sysRlsInfoShow** (pg 129)

Display the release information.

- **moduleShow** (pg 94)

Show the list of all loaded modules.

Disk, memory, interrupt, and TTY commands

Use the commands described in this section to check the hard disk for corruption, and to display information about the devices, the memory, and the filesystem.

- **chkdsk** (pg 56)

Check the hard drive for corruption on the hard drive partitions **/p** and **/u**.

- **devs** (pg 58)
Display all the devices present on the Signaling Server.
- **dosFsConfigShow** (pg 60)
Display information about the DOS file system on the /p partition.
- **iosFdShow** (pg 88)
Display a list of file descriptors presently being used on the Signaling Server.
- **pttyShow** (pg 108)
Display the list of PTTY devices.
- **ttyShow** (pg 131)
Display the attributes of the specified TTY.
- **userMemShow** (pg 132)
Display the User Memory Allocation Map.

Shell commands

This section describes the general commands available in the VxWorks shell.

- **cd** (pg 55)
Change the current default working directory to the specified path.
- **checkStack <task>**
List the task stack sizes and usage for the specified task.
- **copy** (pg 58)
Copy the file specified by <sourceFilename> to the destination file specified by <destFilename>. A value of 0 for sourceFilename or destFilename specifies the standard input or output file.
- **devs** (pg 58)
List devices on Signaling Server.
- **diskFormat <device>**
Format the specified disk.

- **diskInit <device>**
Initialize file system on the specified disk.
- **h** (pg 63)
Display the last 20 commands entered at the VxWorks shell prompt (or set the shell history)
- **help** (pg 81)
Display the VxWorks shell help menu.
- **i** (pg 82)
Display the list of tasks presently running on the system with a summary each associated task control block (TCB).
- **iam <user> <, passwd>**
Set the specified user name and password.
- **lkAddr <address>**
List symbol table entries near the specified address.
- **lkup <string>** (pg 91)
Search the symbol table and list any symbol that contains the string.
- **ll <> or <path>** (pg 92)
List the contents of the current directory with timestamp and size information.
- **ls <path <,long>>**
List the contents of the specified directory <in long format>.
- **mkdir** (pg 93)
Create a new subdirectory named dirName under the current directory.
- **printErrno <value>**
Print the name of a status value.
- **pwd** (pg 109)
Print the current working directory.

- **rename** <old>, <new>
Rename the file specified by <old> to the name specified by <new>.
- **rm** <filename> (pg 109)
Delete the specified file.
- **spyHelp** (pg 123)
Display the VxWorks help menu for the spy functions.
- **ti** <task>
Display the complete task control block for the specified task.
- **version**
Print the VxWorks version information and boot line.
- **whoami**
Print the user name.

Network commands

Use the commands described in this section when working with the network.

- **arpFlush** (pg 54)
Flush all non-permanent entries from the cards ARP cache.
- **arpShow** (pg 54)
Display the current entries in cards system ARP table.
- **icmpstatShow** (pg 84)
Display the statistics of the ICMP protocol.
- **ifShow** (pg 84)
Display the attached network interfaces.
- **inetstatShow** (pg 87)
Display information of all the active IP sockets on the Signaling Server.

- **ipstatShow** (pg 90)
Display the IP protocol statistics.
- **mbufShw** (pg 92)
Display the statistics and the distribution of the low-level buffers used by the IP stack.
- **mRouteAdd <destIP>, <gwIP>, <0xdestNetMask>, <ToS>, 0** (pg 95)
Define multiple routes to the same destination, differentiated by the ToS field and/or the gateway field.
- **mRouteDelete** (pg 95)
Specify the route using the destination address, netmask and ToS.
- **mRouteShow** (pg 96)
Display route information with the ToS bit and mask settings.
- **netHelp** (pg 96)
Display the list of commands used for network information.
- **ping <IPaddress>, <numofpings>** (pg 108)
Send an ICMP ECHO_REQUEST packet to a network host, specified by the parameter IPadr.
- **tcpstatShow** (pg 130)
Display the TCP protocol statistics.
- **udpstatShow** (pg 131)
Display the UDP protocol statistics.

UNISlim trace tool commands

This section describes VxWorks shell commands useful for tracing UNISlim messaging. These commands apply to both the Signaling Server and the VGMC cards.

- **usiLibTraceHelp**
Display all the APIs for the UNISlim trace utility.

- **usiLibTraceSettings**
Display the current trace settings.
- **usiTraceSetOutput <trace output>**
Set the destination output for the trace as specified. Valid <trace output> values are:
1—TTY
2—SYSLOG
- **usiLibTraceOff <IP address>**
Turn off the trace for the specified IP address.
- **usiLibTraceAllOff**
Turn off the trace for all IP addresses.

- **usiLibTraceOn <"IP address">,<SS -> sets filter>,<SS <- sets filter>,<trace output>**

Turn the trace utility on for one telephone where:

<IP address> is the IP address of the telephone,

<SS -> sets filter> is a numeric value representing the type of messages to trace when sent to the telephone,

<SS <- sets filter> is a numeric value representing the type of messages to trace when sent from the telephone, and

<trace output> is a numeric value specifying the destination for the output as follows:

- 1—TTY
- 2—SYSLOG

The valid values for <SS -> sets filter> and <SS <- sets filter> are as follows:

- 0—off
- 1—Broadcast Manager messages
- 2—Audio Manager messages
- 4—Display Manager messages
- 8—Key/Indicator Manager messages
- 16—Basic Manager messages
- 32—Network Manager messages
- 63—All messages

Note: If you wish to track multiple message types, but not all message types, then add together the message type filter values of the ones you wish to track together (i.e. Key and Display = 12). Use the sum as the <SS -> sets filter> or <SS <- sets filter> parameter.

Clearing the Signaling Server to perform a fresh installation

To clear the hard drive, log in to the vxshell and issue the following commands:

-> **ataPartTableClear 0,0**

-> **ataPartTableWrite 0,0**

-> **reboot -1**

The parameters **0,0** for the **ataPartTableClear** and **ataPartTableWrite** commands represent the controller id and the drive id.

A system reboot must be performed after clearing the hard drive.

Command reference for Signaling Server base

This section provides a description of the commands available from the VxWorks shell. These commands are entered at the VxWorks shell prompt (->). The name of the the command is provided in the heading, the syntax with parameters, is given as the first line of text following the title. The commands are listed in alphabetical order.

arpFlush

arpFlush

Flush all non-permanent entries from the cards ARP cache.

arpShow

arpShow

Display the entries in the cards system ARP table.

-> **arpShow**

LINK LEVEL ARP TABLE

destination	gateway	flags	Refcnt	Use	Interface
47.11.254.1	00:e0:16:77:e1:14	405	2	0	fei0
47.11.255.2	00:00:75:45:1e:8f	405	2	297105	fei0
47.11.255.13	00:02:b3:86:2a:a6	405	0	4	lo0
192.168.2.3	00:02:b3:86:2a:a7	405	3	37297	lo0

```

192.168.2.4 00:60:38:bd:0a:ff 405 0 2765 fei1
192.168.2.5 00:60:38:bd:b3:0 405 0 2765 fei1
192.168.2.6 00:60:38:76:0d:6e 405 3 22487 fei1
192.168.2.7 00:60:38:76:0c:4 405 0 6665 fei1
192.168.3.2 00:02:b3:3f:2d:41 405 0 3494266 fei1
192.168.10.2 00:02:b3:65:c1:67 405 1 1022373 fei1

```

value = 75 = 0x4b = 'K'

```

-> arpFlush
value = 0 = 0x0

```

```

-> arpShow

```

LINK LEVEL ARP TABLE

destination	gateway	flags	Refcnt	Use	Interface
47.11.254.1	00:e0:16:77:e1:14	405	2	0	fei0
47.11.255.13	00:02:b3:86:2a:a6	405	0	4	lo0
192.168.2.3	00:02:b3:86:2a:a7	405	3	37297	lo0
192.168.3.2	00:02:b3:3f:2d:41	405	0	3494399	fei1
192.168.10.2	00:02:b3:65:c1:67	405	1	58	fei1

value = 75 = 0x4b = 'K'

cd

cd <path>

Change the current default working directory to the path specified by <path>.

```

-> cd "/u/data"
value = 0 = 0x0

```

```

-> ll

```

size	date	time	name
512	DEC-21-2001	11:13:40	. <DIR>
512	DEC-21-2001	11:13:40	.. <DIR>
11019	DEC-21-2001	11:13:44	GAIN.TBL

```
1468 NOV-21-2002 08:27:30 T_TEMP.TBL
1468 NOV-21-2002 08:27:30 TONE.TBL
716 NOV-21-2002 08:27:30 C_TEMP.TBL
716 NOV-21-2002 08:27:30 CADENCE.TBL
7 MAR-26-2002 18:04:56 REDUIP.TXT
1030 APR-02-2002 12:02:30 ODS.DAT
816 APR-11-2002 15:10:40 ODS.IDX
value = 0 = 0x0
```

```
-> pwd
/u/data
value = 8 = 0x8
```

chkdsk

chkdsk <dev> <, repairLevel>, <entryLenType>

Check the hard drive partition for corruption and print a report of the results, where:

<dev> is the partition to check. Valid values for <dev> are:

- /p—Check the /p partition.
- /u—Check the /p partition.

<, repairLevel> specifies any corrective action to be taken. Valid values for <repairLevel> are:

- 0—Do not write to disk, only report errors found (default value).
- 1—Repair any damage, preserving or salvaging the information in special files created under the root directory of <dev>.
- 2—Do not attempt to salvage any information. Return any clusters with damaged or partial information to the pool of free space.

<entryLenType> specifies the current condition of the disk. All options check the disk, report statistics, and report any errors found. Valid values for <entryLenType> are:

- 0—Disk is not fully used, or average file is more than 16 clusters. This is the usual value used.
- 1—The average file is about 4 clusters in size.
- 2—The disk is full. There are small files averaging 1 cluster in size.

Note 1: All options check the disk, report statistics, and report any errors found.

Note 2: While chkdsk is running, other routines and tasks cannot access the hard disk partition being checked.

-> chkdsk "/p",0,0

Copyright (c) 1993-1996 RST Software Industries Ltd. All rights reserved
ver: 2.6 FCS

Disk Check In Progress ...

total disk space (bytes) :	2,146,467,840
bytes in each allocation unit :	32,768
total allocation units on disk :	65,505
bad allocation units :	0
available bytes on disk :	2,105,966,592
available clusters on disk :	64,269
maximum available contiguous chain (bytes) :	2,094,465,024
available space fragmentation (%) :	1
clusters allocated :	1,236

Done Checking Disk.

value = 0 = 0x0

copy

copy <sourceFilename> <, destFilename>

Copy the specified source file to the specified destination file, where:

<sourceFilename> is the source file to copy, and

<destFilename> is the destination file. If the destination file is not specified, the file is copied to the standard output device (the TTY display).

The copy command can be used to display the contents of text files.

```
-> copy "VERSION.DAT"  
2.00.53  
value = 0 = 0x0
```

```
-> pwd  
/u/config  
value = 10 = 0xa
```

```
-> copy "BOOTP.TAB"  
.subnet1:sm=255.255.254.0:gw=47.11.254.1:ts=192.168.2.2:hn:  
1:tc=.subnet1:ha="00:02:b3:86:2a:a6":ip=47.11.255.13:lp=192.168.2.3  
255.255.0.0 192.168.2.1:to=111:dn=4 0 4 0  
2:tc=.subnet1:ha="00:60:38:8e:2a:f3":ip=47.11.255.17:lp=192.168.2.4  
255.255.0.0 192.168.2.1:to=111:dn=12 0 4 0  
3:tc=.subnet1:ha="00:60:38:bd:b3:01":ip=47.11.255.42:lp=192.168.2.5  
255.255.0.0 192.168.2.1:to=111:dn=16 0 3 0  
value = 0 = 0x0
```

devs

devs

Display all the devices present on the Signaling Server.

The hard drive partitions are **/p** and **/u**, the floppy drive is device **/f0** and the CDROM drive is device **/cd0**.

```
-> devs
drv name
0 /null
1 /dev/rtc
2 /tyCo/0
2 /tyCo/1
3 /aioPipe
7 /bsp
9 nbvws042:
4 /p
10 /vio
11 /tgtsvr
4 /u
12 /cd0
4 /f0
13 /pty/pty00.S
14 /pty/pty00.M
13 /pty/pty01.S
14 /pty/pty01.M
13 /pty/pty02.S
14 /pty/pty02.M
13 /pty/pty03.S
14 /pty/pty03.M
3 /pipe/bootpd
3 /pipe/srv.6
3 /pipe/rudp
3 /pipe/srv.39
15 /locale
3 /pipe/srv.38
4 /ums
3 /pipe/srv.48
3 /pipe/srv.49
4 /webxml/
value = 25 = 0x19
```

dosFsConfigShow

dosFsConfigShow

Display information about the DOS file system on the `/p` partition.

You can use this command to determine the amount of free space available on the hard drive partition.

```
-> dosFsConfigShow
device name:      /p
total number of sectors: 4192902
bytes per sector: 512
media byte:      0xf8
# of sectors per cluster: 64
# of reserved sectors: 1
# of FAT tables: 2
# of sectors per FAT: 256
max # of root dir entries: 512
# of hidden sectors: 63
removable medium: false
disk change w/out warning: not enabled
auto-sync mode:  not enabled
long file names: not enabled
exportable file system: not enabled
lowercase-only filenames: not enabled
volume mode:     O_RDWR (read/write)
available space: 2105966592 bytes
max avail. contig space: 2094465024 bytes
value = 0 = 0x0
->
```

dumptab

dumptab

Display the contents of the BOOTP server database.

The contents displayed by this command match the contents of the `/u/config/bootp.tab` file.

```
-> dumptab
```

```
# main 2.4.3
# (null): dump of bootp server database.
# Dump taken FRI NOV 29 11:06:26 2002
#
# Legend: (see bootptab.5)
# first field -- hostname (not indented)
# bf -- bootfile
# bs -- bootfile size in 512-octet blocks
# cs -- cookie servers
# df -- dump file name
# dn -- domain name
# ds -- domain name servers
# ef -- extension file
# ex -- exec file (YORK_EX_OPTION)
# gw -- gateways
# ha -- hardware address
# hd -- home directory for bootfiles
# hn -- host name set for client
# ht -- hardware type
# im -- impress servers
# ip -- host IP address
# lg -- log servers
# lp -- LPR servers
# ms -- message size
# mw -- min wait (secs)
# ns -- IEN-116 name servers
# nt -- NTP servers (RFC 1129)
# ra -- reply address override
# rl -- resource location protocol servers
# rp -- root path
# sa -- boot server address
# sm -- subnet mask
# sw -- swap server
# tc -- template host (points to similar host entry)
# td -- TFTP directory
# to -- time offset (seconds)
# ts -- time servers
# vm -- vendor magic number
# yd -- YP (NIS) domain
# ys -- YP (NIS) servers
# Tn -- generic option tag n
```

```
1:\
:dn=4 0 4 0:\
:gw=47.11.254.1:\
:hn:\
:ht=1:ha="00:02:B3:86:2A:A6":\
:ip=47.11.255.13:\
:lp=192.168.2.3, 255.255.0.0, 192.168.2.1:\
:sm=255.255.254.0:\
:to=111:\
:ts=192.168.2.2:

2:\
:dn=12 0 4 0:\
:gw=47.11.254.1:\
:hn:\
:ht=1:ha="00:60:38:8E:2A:F3":\
:ip=47.11.255.17:\
:lp=192.168.2.4, 255.255.0.0, 192.168.2.1:\
:sm=255.255.254.0:\
:to=111:\
:ts=192.168.2.2:

3:\
:dn=16 0 3 0:\
:gw=47.11.254.1:\
:hn:\
:ht=1:ha="00:60:38:BD:B3:01":\
:ip=47.11.255.42:\
:lp=192.168.2.5, 255.255.0.0, 192.168.2.1:\
:sm=255.255.254.0:\
:to=111:\
:ts=192.168.2.2:

.subnet1:\
:gw=47.11.254.1:\
:hn:\
:sm=255.255.254.0:\
:ts=192.168.2.2:

value = 29 = 0x1d
```

h

h <n>

Display the last 20 commands entered at the VxWorks shell prompt (->). Historical commands can be recalled and edited.

The VxWorks shell switches to edit history mode when you press the ESC key. The shell exits edit history mode and executes the edited historical command when you press the RETURN key.

The commands available in edit history mode are listed in the following sections.

Note: In the following sections, the default value for the parameter <n> is 1.

Movement and search commands

The commands described in this section provide you with the ability to move around in and search the list of history commands.

For the commands using parameter <n>, the value of <n> must be a valid positive number. If you do not specify a value for <n>, the default value of 1 is used.

- **<n>G**—Go to shell command number <n>.
- **<n>k**—Go back the specified number, <n>, of shell commands.
- **/<s>**—Search for the specified string, <s>, backward in history.
- **?<s>**—Search for the specified string, <s>, forward in history.
- **< n>h**—Move left <n> characters.
- **<n>l** or **<n> “space”**—Move right <n> characters.
- **<n>w**—Move <n> words forward.
- **<n>b**—Move <n> words back.
- **f<c>**—Find character <c>, searching forward in history.
- **F<c>**—Find character <c>, searching backward in history.

- **\$**—Go to end of line.
- **0**—Go to start of line.

Insertion commands

The commands listed in this section are used to insert or change text when editing historical commands. Input is expected until an ESC is pressed.

- **a**—Append.
- **A**—Append at end of line.
- **cl** or **c SPACE**—Change character (deletes a character and enters input mode).
- **cw** —Change word (deletes a word and enters input mode).
- **cc** or **S**—Change the entire line.
- **c\$** or **C**—Change everything from the cursor to the end of line.
- **i**—Insert.
- **I**—Insert at the beginning of the line.
- **R**—Type over characters.

Editing commands

For the commands using parameter <n>, the value of <n> must be a valid positive number. If you do not specify a value for <n>, the default value of 1 is used.

- **<n>r<c>**—Replace the following <n> characters with <c>.
- **<n>x**—Delete <n> characters starting at the cursor.
- **<n>X**—Delete< n> characters to the left of the cursor.
- **dl**—Delete a character.
- **dw**—Delete a word.
- **dd**—Delete the entire line.
- **d\$** or **D**—Delete everything from the cursor to the end of line.
- **p**—Put last deletion after the cursor.

- **P**—Put last deletion before the cursor.
- **u**—Undo last command.

Special commands

- **CTRL-U**—Delete the line and exit edit mode.
- **CTRL+L**—Redraw the line.
- **CTRL+D**—Complete the symbol name.
- **RETURN**—Give the line to the shell and exit edit mode.

```
-> h
339 IPInfoShow
340 lkup "IPInfoShow"
341 pwd
342 cd /u/data
343 cd "/u/data"
344 ll
345 cd "/u/config"
346 ll
347 mkdir "temp"
348 ll
349 lkup "mRouteAdd"
350 mRouteShow
351 mRouteAdd 47.11.216.250 47.11.216.1 0xfffffe0,0,0
352 mRouteAdd "47.11.216.250", "47.11.216.1",0xfffffe0,0,0
353 mRouteShow
354 mRouteDelete "47.11.216.250",0xfffffe0,0
355 mRouteShow
356 i
357 lkup "routeAdd"
358 h
value = 0 = 0x0
```

H323CallTrace ch on

H323CallTrace ch on

Activate H.323 message tracing for all channels.

```
oam> H323CallTrace ch on
oam>
oam>
oam> 11/01/05 15:41:54 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) Q931 setup
11/01/05 15:41:54 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) Q931 callProceeding
11/01/05 15:41:54 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) Q931 alerting
11/01/05 15:41:56 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) Q931 connect
11/01/05 15:41:56 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) H245 terminalCapabilitySet
11/01/05 15:41:56 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) H245 masterSlaveDetermination
11/01/05 15:41:56 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) H245 terminalCapabilitySet
11/01/05 15:41:56 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) H245 terminalCapabilitySetAck

oam> 11/01/05 15:41:56 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) H245 masterSlaveDetermination
11/01/05 15:41:56 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) H245 masterSlaveDeterminationAck
11/01/05 15:41:56 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:41:56 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:41:56 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) H245 terminalCapabilitySetAck
11/01/05 15:41:56 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) H245 masterSlaveDeterminationAck
11/01/05 15:41:56 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:41:58 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) H245 closeLogicalChannel
11/01/05 15:41:58 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:41:58 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) H245 terminalCapabilitySet
11/01/05 15:41:58 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:41:58 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
```

called:4801 remote IP:192.168.19.50(1720) H245 closeLogicalChannel

H323CallTrace ch (single channel)

H323CallTrace ch <channelNum> <MsgRecv> <MsgSend>

Activate or deactivate H.323 message tracing, where:

<channelNum> is the channel number of the VTRK to trace. Values range from 0 - maximum channel number,

<MsgRecv> activates or deactivates H.323 message tracing for messages sent to the specified channels. Values are ON andOFF, and

<MsgSend> activates or deactivates H.323 message tracing for messages sent from the specified channels. Values are ON andOFF.

```
oam> H323CallTrace ch 01 on on
oam>
oam>
oam> 11/01/05 15:45:25 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) Q931 setup
11/01/05 15:45:25 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) Q931 callProceeding
11/01/05 15:45:25 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) Q931 alerting
11/01/05 15:45:27 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) Q931 connect
11/01/05 15:45:27 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) H245 terminalCapabilitySet
11/01/05 15:45:27 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) H245 masterSlaveDetermination
11/01/05 15:45:27 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) H245 terminalCapabilitySet
11/01/05 15:45:27 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) H245 terminalCapabilitySetAck
11/01/05 15:45:27 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) H245 masterSlaveDetermination
```

11/01/05 15:45:27 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 masterSlaveDeterminationAck
11/01/05 15:45:27 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:45:27 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:45:27 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 terminalCapabilitySetAck
11/01/05 15:45:27 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 masterSlaveDeterminationAck
11/01/05 15:45:27 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:45:28 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 closeLogicalChannel
11/01/05 15:45:28 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:45:28 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 terminalCapabilitySet
11/01/05 15:45:28 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:45:28 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 closeLogicalChannel
11/01/05 15:45:28 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:45:28 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 requestChannelClose
11/01/05 15:45:28 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:45:28 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 endSessionCommand
11/01/05 15:45:28 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:45:28 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) Q931 releaseComplete
11/01/05 15:45:28 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 closeLogicalChannelAck
11/01/05 15:45:28 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:45:28 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 closeLogicalChannel
11/01/05 15:45:28 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 terminalCapabilitySetAck
11/01/05 15:45:28 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500

called:4801 remote IP:192.168.19.50(1720) Q931 facility
 11/01/05 15:45:28 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500
 called:4801 remote IP:192.168.19.50(1720) H245 endSessionCommand
 11/01/05 15:45:28 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500
 called:4801 remote IP:192.168.19.50(1720) Q931 facility

H323CallTrace ch (range of channels)

**H323CallTrace ch <start_chNum> <end_chNum> <MsgRecv>
 <MsgSend>**

Activate or deactivate H.323 message tracing for a range of VTRK channels, where:

<start_chNum> is the first channel number in the range of channels. Values range from 0 - maximum channel number.

<end_chNum> is the last channel number in the range of channels. Values range from 0 - maximum channel number, but it must be greater than <start_chNum>.

<MsgRecv> activates or deactivates H.323 message tracing for messages sent to the specified channels. Values are ON andOFF.

<MsgSend> activates or deactivates H.323 message tracing for messages sent from the specified channels. Values are ON andOFF.

```
oam> H323CallTrace ch 01 06 on on
oam>
oam> 11/01/05 15:46:02 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) Q931 setup
11/01/05 15:46:02 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) Q931 callProceeding
11/01/05 15:46:02 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) Q931 alerting
11/01/05 15:46:04 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) Q931 connect
11/01/05 15:46:04 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500
called:4801 remote IP:192.168.19.50(1720) H245 terminalCapabilitySet
```

11/01/05 15:46:04 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 masterSlaveDetermination
11/01/05 15:46:04 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 terminalCapabilitySet
11/01/05 15:46:04 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 terminalCapabilitySetAck
11/01/05 15:46:04 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 masterSlaveDetermination
11/01/05 15:46:04 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 masterSlaveDeterminationAck
11/01/05 15:46:04 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:04 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:04 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 terminalCapabilitySetAck
11/01/05 15:46:04 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 masterSlaveDeterminationAck
11/01/05 15:46:04 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:05 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 closeLogicalChannel
11/01/05 15:46:05 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:05 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 terminalCapabilitySet
11/01/05 15:46:05 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:05 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 closeLogicalChannel
11/01/05 15:46:05 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:05 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 requestChannelClose
11/01/05 15:46:05 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:05 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) H245 endSessionCommand
11/01/05 15:46:05 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:05 LOG0006 NPM: H323CallTrace: Send chid:1 calling:4500 called:4801 remote IP:192.168.19.50(1720) Q931 releaseComplete

11/01/05 15:46:05 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500
 called:4801 remote IP:192.168.19.50(1720) H245 closeLogicalChannelAck
 11/01/05 15:46:05 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500
 called:4801 remote IP:192.168.19.50(1720) Q931 facility
 11/01/05 15:46:05 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500
 called:4801 remote IP:192.168.19.50(1720) H245 closeLogicalChannel
 11/01/05 15:46:05 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500
 called:4801 remote IP:192.168.19.50(1720) H245 terminalCapabilitySetAck
 11/01/05 15:46:05 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500
 called:4801 remote IP:192.168.19.50(1720) Q931 facility
 11/01/05 15:46:05 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500
 called:4801 remote IP:192.168.19.50(1720) H245 endSessionCommand
 11/01/05 15:46:05 LOG0006 NPM: H323CallTrace: Recv chid:1 calling:4500
 called:4801 remote IP:192.168.19.50(1720) Q931 facility
 oam> 11/01/05 15:46:14 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
 called:4500 remote IP:192.168.19.50(1720) Q931 setup
 11/01/05 15:46:14 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801
 called:4500 remote IP:192.168.19.50(1720) Q931 callProceeding
 11/01/05 15:46:14 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801
 called:4500 remote IP:192.168.19.50(1720) Q931 alerting
 11/01/05 15:46:17 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801
 called:4500 remote IP:192.168.19.50(1720) Q931 connect
 11/01/05 15:46:17 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
 called:4500 remote IP:192.168.19.50(1720) H245 terminalCapabilitySet
 11/01/05 15:46:17 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
 called:4500 remote IP:192.168.19.50(1720) H245 masterSlaveDetermination
 11/01/05 15:46:17 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
 called:4500 remote IP:192.168.19.50(1720) Q931 facility
 11/01/05 15:46:17 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801
 called:4500 remote IP:192.168.19.50(1720) H245 terminalCapabilitySet
 11/01/05 15:46:17 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
 called:4500 remote IP:192.168.19.50(1720) H245 terminalCapabilitySetAck
 11/01/05 15:46:17 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801
 called:4500 remote IP:192.168.19.50(1720) H245 masterSlaveDetermination
 11/01/05 15:46:17 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
 called:4500 remote IP:192.168.19.50(1720) H245 masterSlaveDeterminationAck
 11/01/05 15:46:17 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801
 called:4500 remote IP:192.168.19.50(1720) H245 terminalCapabilitySetAck
 11/01/05 15:46:17 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801
 called:4500 remote IP:192.168.19.50(1720) H245 masterSlaveDeterminationAck
 11/01/05 15:46:17 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801
 called:4500 remote IP:192.168.19.50(1720) Q931 facility
 11/01/05 15:46:17 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801

called:4500 remote IP:192.168.19.50(1720) Q931 facility
oam> 11/01/05 15:46:21 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) H245 terminalCapabilitySet
11/01/05 15:46:21 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:21 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) H245 closeLogicalChannel
11/01/05 15:46:21 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:21 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) H245 closeLogicalChannel
11/01/05 15:46:21 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:21 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) H245 requestChannelClose
11/01/05 15:46:21 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:21 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) H245 endSessionCommand
11/01/05 15:46:21 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:21 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) Q931 releaseComplete
11/01/05 15:46:21 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) H245 closeLogicalChannel
11/01/05 15:46:21 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) H245 terminalCapabilitySetAck
11/01/05 15:46:21 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:21 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) H245 closeLogicalChannelAck
11/01/05 15:46:21 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:21 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) H245 endSessionCommand
11/01/05 15:46:21 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) Q931 facility

H323CallTrace num (calling or called party only)

H323CallTrace num <cnum> <MsgRecv> <MsgSend>

Activate or deactivate H.323 message tracing for the specified called or calling number. If tracing is activated for a calling or called number associated with a VTRK session, the H.323 messages associated with the VTRK are traced. The parameters for this command are:

<cnum> is the calling or the called number for which the trace is activated or deactivated. Values for <cnum> can range from 1 to 32 numeric digits and can represent a partial calling or called number.

<MsgRecv> activates or deactivates H.323 message tracing for messages sent to the VTRK associated with <cnum>. Values are ON andOFF.

<MsgSend> activates or deactivates H.323 message tracing for messages sent from the VTRK associated with <cnum>. Values are ON andOFF.

```
oam> H323CallTrace num 4500 on on
oam>
oam> 11/01/05 15:46:39 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) Q931 setup
11/01/05 15:46:39 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) Q931 callProceeding
11/01/05 15:46:40 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) Q931 alerting
11/01/05 15:46:42 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) Q931 connect
11/01/05 15:46:42 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) H245 terminalCapabilitySet
11/01/05 15:46:42 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) H245 masterSlaveDetermination
11/01/05 15:46:42 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:42 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) H245 terminalCapabilitySet
11/01/05 15:46:42 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) H245 terminalCapabilitySetAck
11/01/05 15:46:42 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) H245 masterSlaveDetermination
```

11/01/05 15:46:42 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801 called:4500 remote IP:192.168.19.50(1720) H245 masterSlaveDeterminationAck
11/01/05 15:46:42 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801 called:4500 remote IP:192.168.19.50(1720) H245 terminalCapabilitySetAck
11/01/05 15:46:42 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801 called:4500 remote IP:192.168.19.50(1720) H245 masterSlaveDeterminationAck
11/01/05 15:46:42 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801 called:4500 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:42 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801 called:4500 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:43 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801 called:4500 remote IP:192.168.19.50(1720) H245 closeLogicalChannel IP:192.168.19.50(1720) H245 closeLogicalChannelAck
11/01/05 15:46:43 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801 called:4500 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:43 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801 called:4500 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:43 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801 called:4500 remote IP:192.168.19.50(1720) H245 terminalCapabilitySet
11/01/05 15:46:43 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801 called:4500 remote IP:192.168.19.50(1720) H245 closeLogicalChannel
11/01/05 15:46:43 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801 called:4500 remote IP:192.168.19.50(1720) H245 terminalCapabilitySetAck
11/01/05 15:46:43 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801 called:4500 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:43 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801 called:4500 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:43 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801 called:4500 remote IP:192.168.19.50(1720) H245 closeLogicalChannel
11/01/05 15:46:43 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801 called:4500 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:43 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801 called:4500 remote IP:192.168.19.50(1720) H245 requestChannelClose
11/01/05 15:46:43 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801 called:4500 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:43 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801 called:4500 remote IP:192.168.19.50(1720) H245 endSessionCommand
11/01/05 15:46:43 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801 called:4500 remote IP:192.168.19.50(1720) H245 endSessionCommand
11/01/05 15:46:43 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801 called:4500 remote IP:192.168.19.50(1720) Q931 facility
11/01/05 15:46:43 LOG0006 NPM: H323CallTrace: Send chid:6 calling:4801 called:4500 remote IP:192.168.19.50(1720) Q931 facility

11/01/05 15:46:43 LOG0006 NPM: H323CallTrace: Recv chid:6 calling:4801
called:4500 remote IP:192.168.19.50(1720) Q931 releaseComplete

H323CallTrace num (with numbering plan or number type)

**H323CallTrace num <cnum> <NPI> <TON> <MsgRecv>
<MsgSend>**

Activate or deactivate H.323 call tracing for VTRK sessions associated with the specified called or calling party number, the specified numbering plan and the specified type of call. The parameters for this command are as follows:

<cnum> is the calling or the called number for which the trace is activated or deactivated. Values for <cnum> can range from 1 to 32 numeric digits and can represent a partial calling or called number.

<NPI> is the numbering plan identifier. Values are:

- 0—ALL NPIs
- 1—Unknown
- 2—ISDN/telephone numbering plan (E.164)
- 3—Private numbering plan
- 4—E.163
- 5—Telex numbering plan
- 6—Data numbering plan

7—National standard numbering plan

<TON> is the type of number involved in the call. Values are:

- 0—All TONs
- 1—Unknown Number
- 2—International Number
- 3—National Number
- 4—Network Specific Number
- 5—Subscriber Number
- 6—L1 Regional Number
- 7—L0 Regional Number

<MsgRecv> activates or deactivates H.323 message tracing for messages sent to the VTRK associated with <cnum>. Values are ON andOFF.

<MsgSend> activates or deactivates H.323 message tracing for messages sent from the VTRK associated with <cnum>. Values are ON andOFF.

H323GwShow

H323GwShow

Display a snapshot summary of the state of the virtual trunk settings.

```
oam> H323GwShow
Npm status:           Active
Active GateKeeper:    192.168.19.51 (primary)
GateKeeper registration status: registered, TTL: 25 secs, re-register: 12 secs
Channels Busy / Idle / Total:  0 / 6 / 6
Stack version:        RadVision 4.1.0.19
Channel tracing:      -1
Signaling Server H323 ID :  SS_N318
```

H323GwShow ch

H323GwShow ch <channelNum>

Display a snapshot summary of the state of the virtual trunk settings and a snapshot of the active call on the specified channel if the call exists, where:

<channelNum> is the channel number to trace. Values range from 0-maximum channel number.

```
oam> H323GwShow ch 01
Npm status:           Active
Active GateKeeper:    192.168.19.51 (primary)
GateKeeper registration status: registered, TTL: 25 secs, re-register: 19 secs
Channels Busy / Idle / Total:  1 / 5 / 6
Stack version:        RadVision 4.1.0.19
Channel tracing:      -1
Signaling Server H323 ID :  SS_N318
```

Chan	Direction	CallState	RxState	TxState	Codec	AirTime	FS	MS
Fax	DestNum	RemoteIP						

```
-----
1 Terminate Connected Connected Connected G_711_u_law_20MS_NOVAD
18 yes m no 4801 192.168.19.50
```

H323GwShow num (calling or called party only)

H323GwShow num <cnum>

Display a snapshot summary of the state of the virtual trunk settings and a snapshot of the active calls associated with the specified calling or called number, where:

<cnum> is the calling or called phone number that triggers the trace. Values for <cnum> range from 1 to 32 numeric digits and it can be a partial calling or called number.

```
oam> H323GwShow num 4500
Npm status:           Active
```

Active GateKeeper: 192.168.19.51 (primary)
GateKeeper registration status: registered, TTL: 25 secs, re-register: 14 secs
Channels Busy / Idle / Total: 0 / 6 / 6
Stack version: RadVision 4.1.0.19
Channel tracing: -1
Signaling Server H323 ID : SS_N318

Calling/Called Party Number: 4500
Numbering Plan Indicator: Undefined
Type Of Number: Undefined
No active calls for the number: 4500, NPI: Undefined, TON: Undefined

H323GwShow num (with numbering plan or number type)

H323GwShow num <cnum> <NPI> <TON>

Display a snapshot summary of the state of the virtual trunk settings and a snapshot of the active calls associated with the specified calling or called

number and having the specified NPI and TON values. The parameters for this command are as follows:

<num> is the calling or called telephone number to be traced. Values for <num> range from 1 to 32 numeric digits. <num> can represent a partial calling or called number.

<NPI > specifies the type of numbering plan that must be employed by the call. Only calls using this type of numbering plan will be traced. Values for <NPI> are:

- 0—ALL NPIs
- 1—Unknown
- 2—ISDN/telephone numbering plan (E.164)
- 3—Private numbering plan
- 4—E.163
- 5—Telex numbering plan
- 6—Data numbering plan
- 7—National standard numbering plan

<TON> specifies the type of number to use as a filter for tracing. Only calls using this TON setting will be traced. Values for <TON> are:

- 0—All TONs
- 1—Unknown Number
- 2—International Number
- 3—National Number
- 4—Network Specific Number
- 5—Subscriber Number
- 6—L1 Regional Number
- 7—L0 Regional Number

```
oam> H323GwShow num 4500 3 7
Npm status:           Active
Active GateKeeper:    192.168.19.51 (primary)
GateKeeper registration status: registered, TTL: 25 secs, re-register: 3 secs
Channels Busy / Idle / Total:  1 / 5 / 6
Stack version:        RadVision 4.1.0.19
Channel tracing:      -1
Signaling Server H323 ID :   SS_N318
```

Calling/Called Party Number: 4500
Numbering Plan Indicator: Private
Type Of Number: L0Regional
Chan Direction CallState RxState TxState Codec AirTime FS MS
Fax DestNum RemoteIP

6 Originate Connected Connected Connected G_711_u_law_20MS_NOVAD
18 yes s no 4500 192.168.19.50

H323Output

H323Output <output_destination> <file_pathname>

Direct the H.323 message trace results to the specified destination, where:

<output_destination> is the desired output destination. Values are:

- 1—TTY
- 2—RPTLOG
- 3—File
- 4—File and TTY

If the value of <output_destination> is 1 or 2, <file_pathname> is not used.

If the value of <output_destination> is 3 or 4, <file_pathname> is required. In this case, <file_pathname> is a string, encapsulated in quotes, that specifies the name of the destination output file.

```
oam> H323Output 3 "Testcap.txt"
```

H323TraceShow

H323TraceShow

Display the trace settings, including the output destination and filename, and all active traces for the H323CallTrace trace tool.

```
oam> H323TraceShow
```


Output to TTY

Calling/called number	NPI	TON	H323MsgRecv	H323MsgSend
4500	0 0	ON	OFF	

Channels	H323MsgRecv (VTRK->NPM)	H323MsgSend (NPM->VTRK)
0 - 382	OFF	OFF

help

help

Display the VxWorks shell help menu.

-> help

- help** Print this list
- dbgHelp** Print debugger help info
- nfsHelp** Print nfs help info
- netHelp** Print network help info
- spyHelp** Print task histogrammer help info
- timexHelp** Print execution timer help info
- h** [n] Print (or set) shell history
- i** [task] Summary of tasks' TCBS
- ti** task Complete info on TCB for task
- sp** adr,args... Spawn a task, pri=100, opt=0, stk=20000
- taskSpawn** name,pri,opt,stk,adr,args... Spawn a task
- td** task Delete a task
- ts** task Suspend a task
- tr** task Resume a task
- d** [adr[,nunits[,width]]] Display memory
- m** adr[,width] Modify memory
- mRegs** [reg[,task]] Modify a task's registers interactively
- pc** [task] Return task's program counter
- version** Print VxWorks version info, and boot line

Type <CR> to continue, Q<CR> to stop:

- iam** "user"[,"passwd"] Set user name and passwd
- whoami** Print user name
- devs** List devices

cd "path" Set current working path
pwd Print working path
ls ["path"][,long] List contents of directory
ll ["path"] List contents of directory - long format
rename "old","new" Change name of file
copy ["in"][,"out"] Copy in file to out file (0 = std in/out)
ld [syms[,noAbort][,"name"]] Load stdin, or file, into memory
(syms = add symbols to table:
-1 = none, 0 = globals, 1 = all)
lkup ["substr"] List symbols in system symbol table
lkAddr address List symbol table entries near address
checkStack [task] List task stack sizes and usage
printErrno value Print the name of a status value
period secs,adr,args... Spawn task to call function periodically
repeat n,adr,args... Spawn task to call function n times
(0=forever)
diskFormat "device" Format disk

diskInit "device" Initialize file system on disk

Type <CR> to continue, Q<CR> to stop:

squeeze "device" Squeeze free space on RT-11 device

NOTE: Arguments specifying 'task' can be either task ID or name.

value = 1 = 0x1
->

i

i

Display the list of tasks running on the system with a summary of each associated task control block (TCB).

The entry procedure, task ID, priority, status, program counter, stack pointer, error number and delay for each task is displayed.

-> i

NAME	ENTRY	TID	PRI	STATUS	PC	SP	ERRNO	DELAY
tExcTask	_excTask	cb68c00	0	PEND	41534e6	cb68b70	3006b	0
tLogTask	_logTask	cb662fc	0	PEND	41534e6	cb66268	0	0
tSysWork	408f7e0	cb73e08	1	DELAY	40a9cbe	cb73dbc	3d0001	25
tShell	_shell	8b1836c	1	READY	40d48b0	8b1802c	c0002	0
tWdbTask	40f18e0	bb59dd0	3	PEND	40a83e8	bb59d28	0	0
tPxTimer	_pxTaskInit	ba24b9c	10	DELAY	40a9cbe	ba24b30	4	3592
tAioIoTask1	_aioIoTask	cb44f30	50	PEND	40a83e8	cb44ed8	c0002	0
tAioIoTask0	_aioIoTask	cb3dd9c	50	PEND	40a83e8	cb3dd44	3d0002	0
tNetTask	_netTask	bd701c4	50	PEND	40a83e8	bd7016c	41	0
tAioWait	_aioWaitTask	cb4c0c4	51	PEND	40a83e8	cb4be08	0	0
tFtpdTask	4022090	bcd230c	55	PEND	40a83e8	bcd2240	0	0
tTftpdTask	_tftpdTask	bccf164	55	PEND	40a83e8	bcecbdc	d0003	0
tSntpsTask	40d7760	b9ee51c	56	PEND	40a83e8	b9ee34c	0	0
tPortmapd	_portmapd	bcd3870	100	PEND	40a83e8	bcd34a0	16	0
tLogin	_taskEntry__	ba73dd8	100	DELAY	40a9cbe	ba73ce4	0	30
tLogin	_taskEntry__	ba6812c	100	DELAY	40a9cbe	ba680a0	1c0001	30
tRLogind	_rlogind	ba64250	100	PEND	40a83e8	ba64178	0	0
tTelnetd	_telnetd	ba638bc	100	PEND	40a83e8	ba637c8	0	0
shell	_taskEntry__	b9d6804	100	PEND	40a83e8	b9d6484	3d0001	0
tTelnets46	_telnetTask	cbffe6c	100	PEND	40a83e8	cbffb14	d0003	0
tLogin	_taskEntry__	8b1fbc4	100	DELAY	40a9cbe	8b1fb38	1c0001	29
tTelnetc46	40917d0	cbfeab4	100	READY	40a834f	cbfe808	0	0
shell	_taskEntry__	8b1d320	100	DELAY	40a9cbe	8b1d160	30065	50
tRDP	_rudpMgrStar	b9ec1ec	120	PEND	40a83e8	b9ebb18	b	0
tPBX	_pbxTcpRecvT	b9ba55c	120	PEND	40a83e8	b9ba408	0	0
tSnmpd	40e7e20	bccc0cc	150	PEND	40a83e8	bccb748	0	0
tbootpd	_cmain	b9f1c74	200	PEND	40a83e8	b9f1480	0	0
tMAM	_mamMain	b9dabec	200	PEND	411cdf8	b9dab14	0	0
tRPCMGMT	_start_ss_se	b9ca754	200	PEND	40a83e8	b9ca394	3d0004	0
tELC	_electTask	b9b83c8	200	PEND	40a83e8	b9b7ce4	11	0
tVTM	22fba70	b958b18	200	PEND	411cdf8	b958a38	4	0
tSET	2296db8	b940158	200	PEND	411cdf8	b94007c	4	0
tCSV	_csvTask	b936014	200	PEND	411cdf8	b935f30	4	0
tTPS	_tpsTask	b92f87c	200	PEND	40a83e8	b92f010	380003	0
tUMS	_umsServerSt	b90ad18	200	PEND	411cdf8	b90aba8	4	0
tUMC	_umsClientSt	b0e726c	200	PEND	411cdf8	b0e718c	3d0004	0
tVTK	_vtrkTask	b0db41c	200	PEND	411cdf8	b0db340	3d0004	0
tNPM	_npmMain	b0c90c4	200	PEND	40a83e8	b0c8c5c	39	0
tGKNPM	_gkNpmMain	996a584	200	READY	40a83e8	996a120	39	0
tGKVONPM	_gkVoMain	9432b64	200	PEND	40a83e8	943295c	3d0004	0

```
tGKDBM _gkDbmMaintM 9997b60 201 PEND 411cdf8 9997a7c 4 0
tOMM _ommMain b9d16f4 250 PEND 411cdf8 b9d1620 380003 0
tVTI _vtiTask b94675c 250 PEND 411cdf8 b946688 3d0004 0
tGKOMM _gkOmmMain 9971170 250 PEND 411cdf8 997109c 4 0
tHTTTPd 26636d8 8d25130 250 PEND+T 40a83e8 8d24d2c 3d0004 1581
tHTTTPd 26636d8 8cd4f9c 250 PEND+T 40a83e8 8cd4b98 3d0004 1580
tHTTTPd 26636d8 8c84e08 250 PEND+T 40a83e8 8c84a04 3d0004 1580
tHTTTPd 26636d8 8c34c74 250 PEND+T 40a83e8 8c34870 3d0004 1579
tRptd 407f588 ba87d74 255 PEND 40a83e8 ba87a0c c0002 0
tfwBk _umsServerFw b90c61c 255 PEND 411cdf8 b90c544 1c0001 0
value = 0 = 0x0
```

icmpstatShow

icmpstatShow

Display the statistics of the ICMP protocol.

```
-> icmpstatShow
```

```
ICMP:
```

```
5 calls to icmp_error
```

```
0 error not generated because old message was icmp
```

```
Output histogram:
```

```
echo reply: 80243
```

```
destination unreachable: 5
```

```
13 messages with bad code fields
```

```
0 message < minimum length
```

```
0 bad checksum
```

```
0 message with bad length
```

```
Input histogram:
```

```
echo reply: 2
```

```
destination unreachable: 287
```

```
routing redirect: 2
```

```
echo: 80243
```

```
80243 message responses generated
```

```
value = 35 = 0x23 = '#'
```

ifShow

ifShow

Display the attached network interfaces.

The parameters configured on the ELAN and TLAN are displayed. The two IP addresses displayed on the fei1 interface identify the IP address of the card's TLAN interface and the IP address of the Node. The internal software loopback interface is identified by lo. The ethernet address (MAC address) is also displayed for each interface.

-> ifShow

fei (unit number 0):

Flags: (0x8063) UP BROADCAST MULTICAST ARP RUNNING

Type: ETHERNET_CSMACD

Internet address: 47.11.254.209

Broadcast address: 47.11.255.255

Netmask 0xff000000 Subnetmask 0xfffffe00

Ethernet address is 00:02:b3:ee:24:7d

Metric is 0

Maximum Transfer Unit size is 1500

783888934 octets received

139986857 octets sent

6883638 packets received

2717242 packets sent

4458927 non-unicast packets received

411 non-unicast packets sent

2424711 unicast packets received

2716831 unicast packets sent

0 input discards

0 input unknown protocols

0 input errors

0 output errors

0 collisions; 0 dropped

lo (unit number 0):

Flags: (0x8069) UP LOOPBACK MULTICAST ARP RUNNING

Type: SOFTWARE_LOOPBACK

Internet address: 127.0.0.1

Netmask 0xff000000 Subnetmask 0xff000000

Internet address: 47.11.239.230

Netmask 0xff000000 Subnetmask 0xfffff000

Metric is 0

Maximum Transfer Unit size is 32768

907637 packets received; 907637 packets sent

0 multicast packets received

0 multicast packets sent

0 input errors; 0 output errors

0 collisions; 0 dropped

fei (unit number 1):

Flags: (0x8063) UP BROADCAST MULTICAST ARP RUNNING

Type: ETHERNET_CSMACD

Internet address: 47.11.239.235

Broadcast address: 47.11.239.255

Netmask 0xff000000 Subnetmask 0xfffff00

Internet address: 47.11.239.230

Broadcast address: 47.11.239.255

Netmask 0xff000000 Subnetmask 0xfffff00

Ethernet address is 00:02:b3:ee:24:7e

Metric is 0

Maximum Transfer Unit size is 1500

686695629 octets received

36541169 octets sent

8940075 packets received

437702 packets sent

8548192 non-unicast packets received

44745 non-unicast packets sent

391883 unicast packets received

392957 unicast packets sent

0 input discards

0 input unknown protocols

0 input errors

0 output errors

0 collisions; 0 dropped

ifTabShow

ifTabShow

Display the contents of the file system configuration table.

-> ifTabShow

device name : fei

device number : 0

device ip : 47.11.255.29

device netmask : 0xfffffe00

device gateway : 47.11.254.1

device name : fei

device number : 1

device ip : 47.11.249.111

device netmask : 0xfffff00

device gateway :

value = 0 = 0x0

inetstatShow

inetstatShow

Display information about the active IP sockets on the Signaling Server.

-> inetstatShow

Active Internet connections (including servers)

PCB	Proto	Recv-Q	Send-Q	Local Address	Foreign Address	(state)
c07a3f4	TCP	0	0	47.11.255.13.23	47.11.181.116.33917	ESTABLISHED
c079f50	TCP	0	0	0.0.0.0.80	0.0.0.0	LISTEN
c079e48	TCP	0	0	192.168.2.3.1720	0.0.0.0	LISTEN
c079cbc	TCP	0	0	192.168.2.2.1720	0.0.0.0	LISTEN
c079818	TCP	0	0	47.11.255.13.1024	47.11.255.2.15000	ESTABLISHED
c07968c	TCP	0	0	0.0.0.0.1009	0.0.0.0	LISTEN
c0792f0	TCP	0	0	0.0.0.0.23	0.0.0.0	LISTEN
c07926c	TCP	0	0	0.0.0.0.513	0.0.0.0	LISTEN
c079164	TCP	0	0	0.0.0.0.111	0.0.0.0	LISTEN
c078e4c	TCP	0	0	0.0.0.0.21	0.0.0.0	LISTEN
c079ecc	UDP	0	0	0.0.0.0.5000	0.0.0.0	
c079dc4	UDP	0	0	192.168.2.3.1718	0.0.0.0	
c079d40	UDP	0	0	192.168.2.3.1719	0.0.0.0	
c079c38	UDP	0	0	192.168.2.2.1719	0.0.0.0	
c079bb4	UDP	0	0	192.168.2.3.5100	0.0.0.0	
c079b30	UDP	0	0	192.168.2.2.4100	0.0.0.0	
c079aac	UDP	0	0	0.0.0.0.16540	0.0.0.0	
c079920	UDP	0	0	192.168.2.3.7300	0.0.0.0	
c07989c	UDP	0	0	0.0.0.0.16550	0.0.0.0	
c079794	UDP	0	0	47.11.255.13.15000	0.0.0.0	
c079710	UDP	0	0	0.0.0.0.15001	0.0.0.0	
c079500	UDP	0	0	0.0.0.0.20111	0.0.0.0	
c0793f8	UDP	0	0	0.0.0.0.67	0.0.0.0	
c079374	UDP	0	0	0.0.0.0.162	0.0.0.0	
c0791e8	UDP	0	0	47.11.255.13.161	0.0.0.0	
c0790e0	UDP	0	0	0.0.0.0.111	0.0.0.0	
c07905c	UDP	0	0	0.0.0.0.69	0.0.0.0	

value = 1 = 0x1

iosFdShow

iosFdShow

Display a list of the file descriptors in use on the Signaling Server.

```
-> iosFdShow
fd name          drv
3 /tyCo/0        2
4 /aioPipe       3
5 /tyCo/1        2
6 (socket)       8
7 (socket)       8
8 (socket)       8
9 (socket)       8
10 (socket)      8
11 /u/rpt        4
12 /u/rpt/LOG00034.RPT 4
13 /u/rpt        4
14 /u/rpt/LOG00028.RPT 4
15 (socket)      8
16 (socket)      8
17 (socket)      8
18 (socket)      8
19 /pipe/bootpd  3
20 /pipe/srv.6   3
21 (socket)      8
22 /pipe/rudp    3
23 (socket)      8
24 (socket)      8
25 (socket)      8
26 (socket)      8
27 (socket)      8
28 /pipe/srv.39  3
29 (socket)      8
30 (socket)      8
31 /pipe/srv.38  3
32 (socket)      8
33 (socket)      8
34 /pipe/srv.48  3
35 (socket)      8
36 (socket)      8
37 /pipe/srv.49  3
```



```

38 (socket)      8
39 (socket)      8
40 (socket)      8
41 (socket)      8
42 (socket)      8
46 (socket)      8
47 /pty/pty01.M  14
48 /pty/pty01.S  13 in out err
value = 32752 = 0x7ff0
    
```

IPInfoShow

IPInfoShow

Display the IP information.

```

oam> IPInfoShow
Maintenance Interface = fei0
Maintenance IP address = 47.11.254.209
Maintenance subnet mask = 255.255.254.0
Voice Interface = fei1
Voice IP address = 47.11.239.235
Voice subnet mask   = 255.255.255.0
    
```

ROUTE NET TABLE

Destination	Gateway	Flags	Refcnt	Use	Interface
0.0.0.0	47.11.239.1	0x3	1	2341	fei1
47.11.239.0	47.11.239.235	0x101	0	0	fei1
47.11.254.0	47.11.254.209	0x101	0	0	fei0

ROUTE HOST TABLE

Destination	Gateway	Flags	Refcnt	Use	Interface
47.11.239.230	47.11.239.230	0x5	1	25057	lo0
127.0.0.1	127.0.0.1	0x5	3	186	lo0

ipstatShow

ipstatShow

Display the IP protocol statistics.

```
-> ipstatShow
    total 5148667
    badsum 0
    tooshort 0
    toosmall 0
    badhlen 0
    badlen 0
    infragments 0
    fragdropped 0
    fragtimeout 0
    forward 0
    cantforward 170
    redirectsent 0
    unknownprotocol 289
    nobuffers 0
    reassembled 0
    outfragments 0
    noroute 0

value = 1 = 0x1
```

isetShow

isetShow <> or <IP> or <TN>

Display general information about registered telephones where:

<> no parameter specifies that information about all of the registered telephones is displayed,

<IP> specifies that information about the registered telephone with the specified IP address is displayed, and

<TN> specifies that information about the registered telephone with the specified TN is displayed.

oam> isetShow

Set Information

IP Address HWID	NAT Type FWVsn	RegType UNIStimVsn	State SrcPort	Up Time DstPort	Set-TN	Regd-TN
47.11.239.237 096-00-00-00	i2004 18-000ae40684cc-6600	Branch 0602B65	online	0 00:08:28	096-00-00-00	5000
47.11.239.238 096-00-00-01	i2004 18-000ae40684c3-6600	Branch 0602B65	online	0 00:07:37	096-00-00-01	5000

Total sets = 2

lkup

lkup <string>

Search the symbol table and display any symbol containing the specified string. The search string is case-sensitive.

This command is useful for looking up commands.

->lkup "dump"

pppdump	0x040ca510	text	(mainos.sym) (local)
bootpd_dump	0x022811e0	data	(share.sym) (local)
dump_generic	0x022145c0	text	(share.sym) (local)
dumptab	0x02213da0	text	(share.sym)
dumpgkpv	0x026464b8	data	(gk.sym)
result_dump	0x022066d8	text	(share.sym)
pppdumpb	0x040ca670	text	(mainos.sym) (local)
dump_host	0x02214030	text	(share.sym) (local)
sysctl_dumpentry	0x04145f80	text	(mainos.sym)
dumpH450	0x025683f8	text	(gk.sym)
dumpFS	0x02568428	text	(gk.sym)
value = 0	0x0		

ll

ll <> or <path>

List the contents of the current directory or the specified directory with timestamp and size information.

```
-> ll
  size   date   time   name
-----
  512 SEP-07-2002 13:02:50 .      <DIR>
  512 SEP-07-2002 13:02:50 ..     <DIR>
    8 SEP-07-2002 13:02:56 VERSION.DAT
value = 0 = 0x0
```

mbufShw

mbufShw

Display the statistics and the distribution of the low-level buffers used by the IP stack.

```
-> mbufShow
type   number
-----
FREE   : 36858
DATA   : 4
HEADER : 2
SOCKET : 0
PCB    : 0
RTABLE : 0
HTABLE : 0
ATABLE : 0
SONAME : 0
ZOMBIE : 0
SOOPTS : 0
FTABLE : 0
RIGHTS : 0
IFADDR : 0
CONTROL : 0
OOBDATA : 0
IPMOPTS : 0
```

```

IPMADDR : 0
IFMADDR : 0
MRTABLE : 0
TOTAL : 36864
number of mbufs: 36864
number of times failed to find space: 0
number of times waited for space: 0
number of times drained protocols for space: 0
    
```

CLUSTER POOL TABLE

size	clusters	free	usage
64	4096	4092	2496668
128	4096	4096	12299053
256	4096	4096	90049
512	4096	4096	102
1024	1024	1024	4195
2048	1024	1024	0

value = 80 = 0x50 = 'P'

mkdir

mkdir <dirName>

Create a new subdirectory named <dirName> under the current directory.

```

-> ll
  size      date      time      name
-----
  512 DEC-18-2001 16:17:34 .          <DIR>
  512 DEC-18-2001 16:17:34 ..         <DIR>
  130 DEC-18-2001 16:25:34 NET.INI
 1108 AUG-23-2002 20:03:18 CONFIG.INI
   395 AUG-23-2002 19:17:02 BOOTP.TAB
   995 MAY-12-2002 12:16:22 CONFIG.BAK
   395 AUG-23-2002 19:19:06 BOOTP.BAK
   657 NOV-20-2002 17:27:10 UMS.INI
 10024 NOV-20-2002 17:27:18 CONFIG.VAL
  1050 JUL-22-2002 10:11:06 CONFIG.GKP
    28 NOV-20-2002 17:27:10 USERDATA.INI
    
```

value = 0 = 0x0

-> mkdir "temp"

value = 0 = 0x0

-> ll

size	date	time	name	
512	DEC-18-2001	16:17:34	.	<DIR>
512	DEC-18-2001	16:17:34	..	<DIR>
130	DEC-18-2001	16:25:34	NET.INI	
1108	AUG-23-2002	20:03:18	CONFIG.INI	
395	AUG-23-2002	19:17:02	BOOTP.TAB	
995	MAY-12-2002	12:16:22	CONFIG.BAK	
395	AUG-23-2002	19:19:06	BOOTP.BAK	
657	NOV-20-2002	17:27:10	UMS.INI	
10024	NOV-20-2002	17:27:18	CONFIG.VAL	
1050	JUL-22-2002	10:11:06	CONFIG.GKP	
28	NOV-20-2002	17:27:10	USERDATA.INI	
512	NOV-29-2002	16:42:32	TEMP	<DIR>

value = 0 = 0x0

moduleShow

moduleShow

Display the list of all loaded modules.

pdt> moduleShow

MODULE NAME	MODULE ID	GROUP #	TEXT START	DATA START	BSS START
share.sym	0xdd519f0	2	0	0	0
line.sym	0xdd517cc	3	0	0	0
trunk.sym	0xdd515a8	4	0	0	0
gk.sym	0xdd35b90	5	0	0	0
web.sym	0xdd3596c	6	0	0	0

mRouteAdd

mRouteAdd <destIP>, <gwIP>, <0xdestNetMask>, <ToS>, 0

Configure multiple routes to the same destination, differentiated by the <ToS> field and/or the <gwIPaddr> field where:

<destIP> is the destination IP address in dotted notation,

<gwIP> is the gateway IP address in dotted notation,

<0xdestNetMask> is the net mask of the destination in hexadecimal, and

<ToS> is the type of service for this route.

Note: This change is not permanent; rebooting the Signaling Server rebuilds the routing table from the data in the CONFIG.INI file.

```
-> mRouteAdd "47.11.216.250", "47.11.216.1",0xfffffe0,0,0  
value = 0 = 0x0
```

mRouteDelete

mRouteDelete <destIP>, <0xdestNetMask>, <ToS>

Delete the route with the specified destination address, netmask and ToS where:

<destIP> is the destination IP address,

<0xdestNetMask> is the destination net mask in hexadecimal, and

<ToS> is the Type of Service for this route.

Note: This change is not permanent; rebooting the Signaling Server rebuilds the routing table with data from the CONFIG.INI file.

```
-> mRouteDelete "47.11.216.250",0xfffffe0,0
```

value = 0 = 0x0

mRouteShow

mRouteShow

Display the host and network routing tables as well as the ToS bit and mask settings. This command is similar to **routeShow**.

-> mRouteShow

Destination	Mask	TOS	Gateway	Flags	RefCnt	Use	Interface	Proto
0.0.0.0	0	0	192.168.2.1	3	1	3974	fei1	0
47.11.180.0	fffffe00	0	47.11.254.1	3	1	83761	fei0	0
47.11.228.0	fffffe00	0	47.11.254.1	3	1	158	fei0	0
47.11.254.0	fffffe00	0	47.11.255.13	101	0	0	fei0	0
127.0.0.1	0	0	127.0.0.1	5	0	0	lo0	0
192.168.0.0	ffff0000	0	192.168.2.3	101	0	0	fei1	0
192.168.2.2	0	0	192.168.2.2	5	0	7311	lo0	0

value = 0 = 0x0

netHelp

netHelp

Display the commands that provide network information.

-> netHelp

hostAdd "hostname","inetaddr" - add a host to remote host table;
**"inetaddr" must be in standard
 Internet address format e.g.**

"90.0.0.4"

hostShow - print current remote host table

netDevCreate "devname","hostname",protocol
 - create an I/O device to access

files

**on the specified host
 (protocol 0=rsh, 1=ftp)**

routeAdd "destaddr","gateaddr" - add route to route table

routeDelete "destaddr","gateaddr" - delete route from route table

routeShow - print current route table
iam "usr"[,"passwd"] - specify the user name by which you
will be known to remote hosts
(and optional password)
whoami - print the current remote ID
rlogin "host" - log in to a remote host;
"host" can be inet address or
host name in remote host table

Type <CR> to continue, Q<CR> to stop:

ifShow ["ifname"] - show info about network interfaces
inetstatShow - show all Internet protocol sockets
tcpstatShow - show statistics for TCP
udpstatShow - show statistics for UDP
ipstatShow - show statistics for IP
icmpstatShow - show statistics for ICMP
arptabShow - show a list of known ARP entries
mbufShow - show mbuf statistics

EXAMPLE: -> hostAdd "wrs", "90.0.0.2"
-> netDevCreate "wrs:", "wrs", 0
-> iam "fred"
-> copy <wrs:/etc/passwd /* copy file from host "wrs" */
-> rlogin "wrs" /* rlogin to host "wrs" */

value = 1 = 0x1

nrsCollaboratingServerQuery

nrsCollaboratingServerQuery <srvrIP>, <DBselector>

Query one collaborating server from the database where:

<srvrIP> is the IP address of the server to query, and

<DBselector> identifies the database to query. Values are:

- 0—query the active database and
- 1—query the standby database.

nrsCollaboratingServerShow

nrsCollaboratingServerShow <DBselector>

List all the collaborating servers in the specified database where:

<DBselector> identifies the database to query. Values are:

0—query the active database and

1—query the standby database.

```
pd> nrsCollaboratingServerShow 0
```

```
Active DB
```

```
ID ALIASNAME FQDNTYPE SERVERFQDN H323SUPPORT  
RASPORT SIPSUPPORT SIPTRANSPORT SIPPORT NCSSUPPORT  
NCSTRANSPORT NCSPORT PARENTL1DOMAIN PARENTL0DOMAIN
```

```
-----  
-----  
-----
```

```
Total Rows:0
```

nrsDbCommit

nrsDbCommit

Copy the table from the active to the standby database.

Note: The command **nrsDbCutover** must be executed prior to executing this command.

nrsDbCommitNow

nrsDbCommitNow

Perform the cutover and commit at once.

nrsDbCutover

nrsDbCutover

Activate the standby database.

nrsDbRevert

nrsDbRevert

Activate the last committed database.

nrsDbRollback

nrsDbRollback

Copy the table from the active to the standby database.

nrsDBShow

nrsDBShow

Display the status of the primary and alternative NRS and the local NRS database.

```

pdt> nrsDBShow
Local NRS IP Address:      47.11.239.235
Local NRS Role:           Primary NRS
Primary NRS Node PRI_47_11_254_235: 47.11.254.235 ACTIVE
Alternate NRS not configured
AlternatePermanentInService:  OFF
    
```

nrsDefaultRouteQuery

nrsDefaultRouteQuery <endpoint_name> <DBselector>

List the default routes that belong to the specified endpoint in the specified database, where:

<endpoint_name> is the name of the endpoint, and

<DBselector> specifies the database. Values are:

0—query the active database

1—query the standby database.

nrsGKTestQuery

nrsGKTestQuery <DN>, <DN_type>, <origEP>, <DBselector>

Query the H.323 routes on the active or standby NRS database, where:

<origEP> is the name of the originating endpoint,

<DBSelector> specifies the database. Values are:

0—query the active database

1—query the standby database.

nrsGWEndpointQuery

nrsGWEndpointQuery <endpoint_name>, <DBselector>

Query the specified gateway endpoint in the specified database, where:

<endpoint_name> is the name of the endpoint, and

<DBSelector> specifies the database. Values are:

0—query the active database

1—query the standby database.

nrsGWEndpointShow

nrsGWEndpointShow <DBselector>

List the gateway endpoints in the specified database, where:

<DBSelector> specifies the database. Values are:

- 0—query the active database
- 1—query the standby database.

```
oam> nrsGWEndpointShow 0
```

```
Active DB
```

```
ID  ENDPOINTNAME  DESCRIPTION  TANDEMENDPOINT
E164COUNTRYCODE E164AREACODE  INTDIALINGACCESSCODE
L1DDIALINGACCESSCODE  H323SUPPORTTYPE SIPSUPPORTTYPE
AUTHENABLED  PASSWORD  NCSENABLED  FQDNTYPE
SHOSTFQDN  H323TRANSPORT H323PORTSIPTRANSPORT SIP-
PORT PARENTL0DOMAIN NATDIALINGACCESSCODE  LOCALDIAL-
INGACCESSCODE SPECIALNUMBER1 SPECIALNUMBER2
```

```
-----
1  FS_Buffy_1      0                1  2  2        1  0
47.11.239.235 01719 0  5060 1
```

nrsL0DomainQuery

nrsL0DomainQuery <domname>, <L1_domname>, <DBselector>

Query one Level 0 Domain from the specified database, where:

<domname> is the name of the L0 domain,

<L1_domname> is the name of the L1 domain, and

<DBSelector> specifies the database. Values are:

- 0—query the active database
- 1—query the standby database.

Total Rows:1

nrsL1DomainQuery

nrsL1DomainQuery <domname>, <DBselector>

Query one Level 1 Domain from the database, where:

<domname> is the name of the L1 domain, and

<DBSelector> specifies the database. Values are:

0—query the active database

1—query the standby database.

oam> nrsL1DomainQuery FS_System, 0

Identification: 1

Domain name: FS_System

Level 1Domain Description:

L1 Domain Authentication Enabled(0 - off, 1 - on, 2 - not configured): 0

Authentication Password:

E164 Country Code:

E164 Area Code:

International Dialing Access Code:

Level 1 Domain Dialing Access Code:

Parent service domain ID: 1

National Dialing Access Code:

Local Dialing Access Code:

Special Number 1:

Special Number 2:

nrsL1DomainShow

nrsL1DomainShow <DBselector>

List the Level 1 Domains in the database, where:

<DBSelector> specifies the database. Values are:

0—query the active database

1—query the standby database.

oam> nrsL1DomainShow 0

Active DB

**ID DOMAINNAME DESCRIPTION AUTHENABLED PASSWORD
E164COUNTRYCODE E164AREACODE INTDIALINGACCESSCODE
L1DDIALINGACCESSCODE PARENTSERVICEDOMAIN NATDIALIN-
GACCESSCODE LOCALDIALINGACCESSCODE SPECIALNUMBER1
SPECIALNUMBER2**

1 FS_System 0 1

Total Rows:1

nrsRoutingEntryShow

nrsRoutingEntryShow <DBselector>

List the routing entries in the database, where:

<DBSelector> specifies the database. Values are:

- 0—query the active database
- 1—query the standby database.

nrsServiceDomainQuery

nrsServiceDomainQuery <domname>, <DBselector>

Query one service domain from the specified database, where:

<domname> is the service domain name, and

<DBSelector> specifies the database. Values are:

- 0—query the active database
- 1—query the standby database.

oam> nrsServiceDomainQuery BVW_Large_System_LAB 0

Identification: 1

Domain name: BVW_Large_System_LAB
Service Domain Description:

nrsServiceDomainShow

nrsServiceDomainShow <DBselector>

List the service domains in the specified database, where:

<DBSelector> specifies the database. Values are:

- 0—query the active database
- 1—query the standby database.

oam> nrsServiceDomainShow 0

Active DB

ID	DOMAINNAME	DESCRIPTION
1	BVW_Large_System_LAB	

Total Rows:1

nrsSIPTestQuery

nrsSIPTestQuery <TermDomain>, <TermPhoneContext>, <TermDN>, <OrigIP>, <IPType>, <DBSelector>

Query the SIP routes on the specified NRS database, where:

<TermDomain> is the service domain name of the terminating end,

<TermPhoneContext> is the phone context (L0Domain.L1Domain) of the terminating end,

<TermDN> is the DN of the terminating end,

<OrigIP> is the IP address of the originating end,

<IPType> is the IP address type of the originating end. Values are:

- 0—IP4 (only type supported currently)
- 1—IP6
- 2—domain name, and

<DBSelector> specifies the database. Values are:

- 0—query the active database
- 1—query the standby database.

nrsUserEPQuery

nrsUserEPQuery <Service_DomainName>, <endpoint_name>, <DBselector>

Query one user endpoint from the specified database, where:

<DBSelector> specifies the database. Values are:

- 0—query the active database
- 1—query the standby database.

nrsUserEPShow

nrsUserEPShow <DBselector>

List the user endpoints in the specified database, where:

<DBSelector> specifies the database. Values are:

0—query the active database

1—query the standby database.

pbxLinkShow

pbxLinkShow

Display the PBX link status.

```
oam> pbxLinkShow
Active Call Server type = CS 1000E
Active Call Server S/W Release = 400T
Supported Features: CorpDir UserKeyLabel VirtualOffice UseCSPwd I2001
I2004 Ph2 I2002 Ph2 PD/RL/CL QoS Monitoring NAT Traversal
Call Server Main: ip = 47.11.254.35, ConnectID = 0x1ca97b20, BroadcastID =
0x1ca97a20, Link is up
Call Server Redundant: ip = 47.11.254.36, ConnectID = 0x1ca97c20, Broad-
castID = 0x0, Link is down
Call Server Signaling Port = 15000
Call Server Broadcast Port = 15001
Broadcast PortID = 0x1ca39da0
RUDP portID = 0x1ca39e20
Tcp Link state = up
Tcp Signaling Port: 15000
Tcp socket fd: 27
Tcp msgs sent: 236
Tcp msgs recd: 3481
```

ping

ping <IPAddress>, <numofpings>

Send a specified number of ICMP ECHO_REQUEST packets to a specified network host, where:

<IPAddress> specifies the destination network host, and

<numofpings> specifies the number of packets to be sent.

If no value is specified for <numofpings>, the packets continue to be sent until CTRL+C is entered.

The destination network host responds to the request. If a response is not received in less than 5 seconds, the sender times out.

```
-> ping "47.11.181.116",5
PING 47.11.181.116: 56 data bytes
64 bytes from 47.11.181.116: icmp_seq=0. time=0. ms
64 bytes from 47.11.181.116: icmp_seq=1. time=0. ms
64 bytes from 47.11.181.116: icmp_seq=2. time=0. ms
64 bytes from 47.11.181.116: icmp_seq=3. time=0. ms
64 bytes from 47.11.181.116: icmp_seq=4. time=0. ms
----47.11.181.116 PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/0/0
value = 0 = 0x0
```

pttyShow

pttyShow

Display the list of PTTY devices.

```
-> pttyShow
ptyMasterName: /pty/pty00.M
ptySlaveName : /pty/pty00.S
ptyMasterFd : 36
ptySlaveFd : 37
```

```
ptyMasterName: /pty/pty01.M
ptySlaveName : /pty/pty01.S
ptyMasterFd   : 39
ptySlaveFd    : 40
```

```
ptyMasterName: /pty/pty02.M
ptySlaveName : /pty/pty02.S
ptyMasterFd   : 42
ptySlaveFd    : 43
```

```
ptyMasterName:
ptySlaveName :
ptyMasterFd   : -1
ptySlaveFd    : -1
```

value = 19 = 0x13

pwd

pwd

Display the current working directory.

```
-> pwd
/p/data
value = 8 = 0x8
```

rm

rm <filename>

Delete the specified file.

```
-> ll
size      date      time      name
-----
  512 SEP-07-2002 13:02:50 .          <DIR>
  512 SEP-07-2002 13:02:50 ..         <DIR>
    8 SEP-07-2002 13:02:56 VERSION.DAT
  512 NOV-27-2002 14:27:04 TEMP        <DIR>
value = 0 = 0x0
```

```
-> rm "temp"  
value = 0 = 0x0
```

```
-> ll  
size      date      time      name  
-----  
  512 SEP-07-2002 13:02:50 .      <DIR>  
  512 SEP-07-2002 13:02:50 ..     <DIR>  
    8 SEP-07-2002 13:02:56 VERSION.DAT  
value = 0 = 0x0
```

SIPCallTrace

SIPCallTrace <trace_state>

Activate or deactivate SIP tracing for all incoming and outgoing messages on all channels. Values for <trace_state> are ON and OFF.

SIPCallTrace ch (single channel)

SIPCallTrace ch <channelNum> <MsgRecv> <MsgSend>

Activate or deactivate SIP tracing for the specified channel, where:

<channelNum> is the channel number. Values range from 0 - maximum channel number,

<MsgRecv> specifies activation or deactivation of tracing for SIP messages sent to the specified channel. Values are ON and OFF.

<MsgSend> specifies activation or deactivation of tracing for SIP messages sent from the specified channel. Values are ON and OFF.

```
oam> SIPCallTrace ch 033 on on  
oam> 11/01/05 15:22:11 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:22:11 Recv  
chid:33 ip:192.168.19.50:5060 SIP INVITE
```

11/01/05 15:22:11 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:22:11 Send
chid:33 ip:192.168.19.51:5060 SIP response 100

11/01/05 15:22:11 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:22:11 Send
chid:33 ip:192.168.19.51:5060 SIP response 180

11/01/05 15:22:11 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:22:11 Recv
chid:33 ip:192.168.19.50:5060 SIP method PRACK(7)

11/01/05 15:22:11 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:22:11 Send
chid:33 ip:192.168.19.51:5060 SIP response 200

11/01/05 15:22:19 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:22:19 Send
chid:33 ip:192.168.19.51:5060 SIP response 200

11/01/05 15:22:19 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:22:19 Recv
chid:33 ip:192.168.19.50:5060 SIP method ACK(1)

11/01/05 15:22:19 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:22:19 Recv
chid:33 ip:192.168.19.50:5060 SIP method other/unknown(6)

11/01/05 15:22:19 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:22:19 Send
chid:33 ip:192.168.19.51:5060 SIP response 200

11/01/05 15:22:23 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:22:23 Send
chid:33 ip:192.168.19.51:5060 SIP method BYE(2)

11/01/05 15:22:23 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:22:23 Recv
chid:-1 ip:192.168.19.50:5060 SIP response 200

11/01/05 15:22:33 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:22:33 Recv
chid:33 ip:192.168.19.50:5060 SIP INVITE

11/01/05 15:22:33 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:22:33 Send
chid:33 ip:192.168.19.51:5060 SIP response 100

11/01/05 15:22:33 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:22:33 Send
chid:33 ip:192.168.19.51:5060 SIP response 180

11/01/05 15:22:33 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:22:33 Recv
chid:33 ip:192.168.19.50:5060 SIP method PRACK(7)

11/01/05 15:22:33 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:22:33 Send

chid:33 ip:192.168.19.51:5060 SIP response 200

11/01/05 15:22:36 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:22:36 Send
chid:33 ip:192.168.19.51:5060 SIP response 200

11/01/05 15:22:36 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:22:36 Recv
chid:33 ip:192.168.19.50:5060 SIP method ACK(1)

11/01/05 15:22:36 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:22:36 Recv
chid:33 ip:192.168.19.50:5060 SIP method other/unknown(6)

11/01/05 15:22:36 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:22:36 Send
chid:33 ip:192.168.19.51:5060 SIP response 200

SIPCallTrace ch (range of channels)

**SIPCallTrace ch <start_chNum> <end_chNum> <MsgRecv>
<MsgSend>**

Activate or deactivate SIP tracing on a range of VTRK channels, where:

<start_chNum> is the first channel number in range. Values range from 0 - maximum channel number.

<end_chNum> is the last channel number in the range. Values range from 0 - maximum channel number, but it must be greater than <start_chNum>.

<MsgRecv> specifies activation or deactivation of tracing for SIP messages sent to the specified range of channels. Values are ON and OFF.

<MsgSend> specifies activation or deactivation of tracing for SIP messages sent from the specified range of channels. Values are ON and OFF.

oam> SIPCallTrace ch 033 38 on on

**oam> 11/01/05 15:23:30 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:23:30
Send chid:33 ip:192.168.19.51:5060 SIP method BYE(2)**

11/01/05 15:23:30 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:23:30 Recv

chid:-1 ip:192.168.19.50:5060 SIP response 200

11/01/05 15:23:40 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:23:40 Recv
chid:33 ip:192.168.19.50:5060 SIP INVITE

11/01/05 15:23:40 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:23:40 Send
chid:33 ip:192.168.19.51:5060 SIP response 100

11/01/05 15:23:40 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:23:40 Send
chid:33 ip:192.168.19.51:5060 SIP response 180

11/01/05 15:23:40 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:23:40 Recv
chid:33 ip:192.168.19.50:5060 SIP method PRACK(7)

11/01/05 15:23:40 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:23:40 Send
chid:33 ip:192.168.19.51:5060 SIP response 200

oam> 11/01/05 15:24:00 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:24:0 Recv
chid:33 ip:192.168.19.50:5060 SIP method CANCEL(8)

11/01/05 15:24:00 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:24:0 Send
chid:33 ip:192.168.19.51:5060 SIP response 200

11/01/05 15:24:00 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:24:0 Send
chid:33 ip:192.168.19.51:5060 SIP response 487

SIPCallTrace num (calling or called party only)

SIPCallTrace num <cnum> <MsgRecv> <MsgSend>

Activate or deactivate SIP tracing for the specified called or calling number.
If tracing is activated for a calling or called number associated with a VTRK

session, the SIP messages associated with the VTRK are traced. The parameters for this command are as follows:

<cnum> is the calling or the called number. Values for <cnum> can range from 1 to 32 numeric digits and can represent a partial calling or called number.

<MsgRecv> activates or deactivates SIP tracing for messages sent to the VTRK associated with <cnum>. Values are ON and OFF.

<MsgSend> activates or deactivates SIP tracing for messages sent from the VTRK associated with <cnum>. Values are ON and OFF.

```
oam> SIPCallTrace 5500 on on
oam> 11/01/05 15:19:56 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:19:56
Recv chid:33 ip:192.168.19.50:5060 SIP INVITE

11/01/05 15:19:56 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:19:56 Send
chid:33 ip:192.168.19.51:5060 SIP response 100

11/01/05 15:19:56 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:19:56 Send
chid:33 ip:192.168.19.51:5060 SIP response 180

11/01/05 15:19:56 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:19:56 Recv
chid:33 ip:192.168.19.50:5060 SIP method PRACK(7)

11/01/05 15:19:56 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:19:56 Send
chid:33 ip:192.168.19.51:5060 SIP response 200

11/01/05 15:20:00 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:0 Recv
chid:33 ip:192.168.19.50:5060 SIP method CANCEL(8)

11/01/05 15:20:00 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:0 Send
chid:33 ip:192.168.19.51:5060 SIP response 200

11/01/05 15:20:00 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:0 Send
chid:33 ip:192.168.19.51:5060 SIP response 487

11/01/05 15:20:16 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:16 Recv
chid:33 ip:192.168.19.50:5060 SIP INVITE

11/01/05 15:20:16 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:16 Send
```

chid:33 ip:192.168.19.51:5060 SIP response 100

11/01/05 15:20:17 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:17 Send
chid:33 ip:192.168.19.51:5060 SIP response 180

11/01/05 15:20:17 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:17 Recv
chid:33 ip:192.168.19.50:5060 SIP method PRACK(7)

11/01/05 15:20:17 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:17 Send
chid:33 ip:192.168.19.51:5060 SIP response 200

11/01/05 15:20:21 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:21 Recv
chid:33 ip:192.168.19.50:5060 SIP method CANCEL(8)

11/01/05 15:20:21 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:21 Send
chid:33 ip:192.168.19.51:5060 SIP response 200

11/01/05 15:20:21 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:21 Send
chid:33 ip:192.168.19.51:5060 SIP response 487

11/01/05 15:20:26 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:26 Recv
chid:33 ip:192.168.19.50:5060 SIP INVITE

11/01/05 15:20:26 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:26 Send
chid:33 ip:192.168.19.51:5060 SIP response 100

11/01/05 15:20:26 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:26 Send
chid:33 ip:192.168.19.51:5060 SIP response 180

11/01/05 15:20:26 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:26 Recv
chid:33 ip:192.168.19.50:5060 SIP method PRACK(7)

11/01/05 15:20:26 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:26 Send
chid:33 ip:192.168.19.51:5060 SIP response 200

11/01/05 15:20:30 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:30 Send
chid:33 ip:192.168.19.51:5060 SIP response 200

11/01/05 15:20:30 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:30 Recv
chid:33 ip:192.168.19.50:5060 SIP method ACK(1)

11/01/05 15:20:30 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:30 Recv
chid:33 ip:192.168.19.50:5060 SIP method other/unknown(6)

11/01/05 15:20:30 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:30 Send
chid:33 ip:192.168.19.51:5060 SIP response 200

11/01/05 15:20:33 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:33 Recv
chid:33 ip:192.168.19.50:5060 SIP method BYE(2)

11/01/05 15:20:33 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:20:33 Send
chid:33 ip:192.168.19.51:5060 SIP response 200

SIPCallTrace num (with numbering plan or number type)

SIPCallTrace num <cnum> <NPI> <TON> <MsgRecv> <MsgSend>

Activate or deactivate SIP tracing for VTRK sessions associated with the specified called or calling number, the specified NPI, and the specified TON.

If the called or calling number of a VTRK session matches <cnum>; the numbering plan used by the call matches <NPI>, and the type of number

involved in the call matches <TON>, the SIP messages associated with the VTRK are traced. The parameters for this command are as follows:

<cnum> is the calling or the called number. Values for <cnum> can range from 1 to 32 numeric digits and can represent a partial calling or called number.

<NPI> is the numbering plan identifier. Values are:

- 0—ALL NPIs
- 1—Unknown
- 2—ISDN/telephone numbering plan (E.164)
- 3—Private numbering plan
- 4—E.163
- 5—Telex numbering plan
- 6—Data numbering plan
- 7—National standard numbering plan

<TON> is the type of number involved in the call. Values are:

- 0—All TONs
- 1—Unknown Number
- 2—International Number
- 3—National Number
- 4—Network Specific Number
- 5—Subscriber Number
- 6—L1 Regional Number
- 7—L0 Regional Number

<MsgRecv> activates or deactivates SIP tracing for messages sent to the VTRK. Values are ON and OFF.

<MsgSend> activates or deactivates SIP tracing for messages sent from the VTRK. Values are ON and OFF.

oam> SIPCallTrace num 5500 3 7 on on

**The trace settings for Num: 5500, NPI: Private and TON: CDP were already available as follows:
Number : 5500**

NPI : Undefined
TON : Undefined
MsgRecv: On
MsgSend: On
oam> 11/01/05 15:19:19 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:19:19
Recv chid:33 ip:192.168.19.50:5060 SIP INVITE

11/01/05 15:19:19 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:19:19 Send
chid:33 ip:192.168.19.51:5060 SIP response 100

11/01/05 15:19:19 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:19:19 Send
chid:33 ip:192.168.19.51:5060 SIP response 180

11/01/05 15:19:19 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:19:19 Recv
chid:33 ip:192.168.19.50:5060 SIP method PRACK(7)

11/01/05 15:19:19 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:19:19 Send
chid:33 ip:192.168.19.51:5060 SIP response 200

11/01/05 15:19:21 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:19:21 Send
chid:33 ip:192.168.19.51:5060 SIP response 200

11/01/05 15:19:21 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:19:21 Recv
chid:33 ip:192.168.19.50:5060 SIP method ACK(1)

11/01/05 15:19:21 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:19:21 Recv
chid:33 ip:192.168.19.50:5060 SIP method other/unknown(6)

11/01/05 15:19:21 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:19:21 Send
chid:33 ip:192.168.19.51:5060 SIP response 200

11/01/05 15:19:23 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:19:23 Recv
chid:33 ip:192.168.19.50:5060 SIP method BYE(2)

11/01/05 15:19:23 LOG0006 SIPNPM: SIPCallTrace: 11/1/5 15:19:23 Send
chid:33 ip:192.168.19.51:5060 SIP response 200

SIPGwShow

SIPGwShow

Display a snapshot summary of the state of the virtual trunk settings.

Note 1: This command sets the channel id to -1 by default.

Note 2: This command does not show the virtual trunk status.

```
oam> SIPGwShow
SIPNPM Status      : Active
Primary Proxy IP address : 192.168.19.51
Secondary Proxy IP address : 192.168.19.61
Primary Proxy port    : 5060
Secondary Proxy port  : 5060
Active Proxy        : Primary :Registered
Time To Next Registration : 2013 Seconds
Channels Busy / Idle / Total : 0 / 6 / 6
Stack version       : 3.0.4.7
Channel tracing     : -1
Chan Direction CallState SIPState   MediaState Codec      AirTime
FS Fax DestNum RemoteIP
-----
```

Channel id should be a non-zero value

SIPGwShow ch

SIPGwShow ch <channelNum>

Display a snapshot summary of the state of the virtual trunk settings and a snapshot of the active call on the specified channel if the call exists.

<channelNum> is the channel number. Values range from 0-maximum channel number.

```
oam> SIPGwShow ch 33
SIPNPM Status      : Active
Primary Proxy IP address : 192.168.19.51
Secondary Proxy IP address : 192.168.19.61
Primary Proxy port    : 5060
Secondary Proxy port  : 5060
Active Proxy        : Primary :Registered
Time To Next Registration : 1943 Seconds
Channels Busy / Idle / Total : 1 / 5 / 6
```

```

Stack version      : 3.0.4.7
Channel tracing   : -1
Chan Direction CallState SIPState   MediaState Codec      AirTime
FS Fax DestNum RemoteIP
-----
33 Terminate BUSY   Invite Received SendRecv
G_711_u_law_20MS_NOVAD 61   Yes No 5801 192.168.19.155
    
```

SIPGwShow num (calling or called party only)

SIPGwShow num <cnnum>

Display a snapshot summary of the state of the virtual trunk settings and a snapshot of the active calls using the specified calling, called or partial number, where:

<cnnum> is the calling or called number associated with a VTRK session. The number can also be a partial calling or called number. Values range from 1 to 32 numeric digits.

```

oam> SIPGwShow num 5500
SIPNPM Status      : Active
Primary Proxy IP address : 192.168.19.51
Secondary Proxy IP address : 192.168.19.61
Primary Proxy port    : 5060
Secondary Proxy port   : 5060
Active Proxy         : Primary :Registered
Time To Next Registration : 1583 Seconds
Channels Busy / Idle / Total : 1 / 5 / 6
Stack version        : 3.0.4.7
Channel tracing      : -1
Calling/Called Party Number: 5500
Numbering Plan Indicator: Undefined
Type Of Number: Undefined
Chan Direction CallState SIPState   MediaState Codec      AirTime
FS Fax DestNum RemoteIP
-----
33 Terminate BUSY   Invite Received SendRecv
G_711_u_law_20MS_NOVAD 10   Yes No 5801 192.168.19.155
    
```


SIPGwShow num (with numbering plan or number type)

SIPGwShow num <num> <NPI > <TON>

Display a snapshot summary of the state of the virtual trunk settings and a snapshot of the active calls associated with the specified calling or called number, the specified numbering plan, and the specified type of number. The parameters are as follows:

<num> is the calling or called number associated with a VTRK session. The number can also be a partial calling or called number. Values range from 1 to 32 numeric digits.

<NPI> is the numbering plan identifier. Values are:

- 0—ALL NPIS
- 1—Unknown
- 2—ISDN/telephone numbering plan (E.164)
- 3—Private numbering plan
- 4—E.163
- 5—Telex numbering plan
- 6—Data numbering plan
- 7—National standard numbering plan

<TON> is the type of number involved in the call. Values are:

- 0—All TONs
- 1—Unknown Number
- 2—International Number
- 3—National Number
- 4—Network Specific Number
- 5—Subscriber Number
- 6—L1 Regional Number
- 7—L0 Regional Number

```
oam> SIPGwShow num 5500 3 7
SIPNPM Status      : Active
Primary Proxy IP address : 192.168.19.51
Secondary Proxy IP address : 192.168.19.61
Primary Proxy port    : 5060
```

Secondary Proxy port : 5060
Active Proxy : Primary :Registered
Time To Next Registration : 1524 Seconds
Channels Busy / Idle / Total : 1 / 5 / 6
Stack version : 3.0.4.7
Channel tracing : -1
Calling/Called Party Number: 5500
Numbering Plan Indicator: Private
Type Of Number: CDP
Chan Direction CallState SIPState MediaState Codec AirTime FS
Fax DestNum RemoteIP

33 Terminate BUSY Invite Received SendRecv G_711_u_law_20MS_NOVAD
69 Yes No 5801 192.168.19.155

SIPOutput

SIPOutput <output_destination> <file_pathname>

Direct the SIP trace output to the specified output_destination or the specified file, where:

<output_destination> is the output destination. Values are:

- 1—TTY
- 2—RPTLOG
- 3—File

<file_pathname> is a string, encapsulated by quotes, specifying the destination output file.

Note: If the value of <output_destination> is 1 or 2, <file_pathname> is not used. If the value of <output_destination> is 3, <file_pathname> is required.

SIPTraceShow

SIPTraceShow

Display the trace settings, the output destination, the output destination filename, if applicable, and all active traces for the SIPCallTrace trace tool.

spyHelp

spyHelp

Display the VxWorks help menu for the spy functions.

The VxWorks spy function displays the task activity (real time usage) by monitoring the tasks and printing a summary at the specified interval. This function provides a method to determine if one task is using a large amount of the CPU resources. The monitoring and report printing use some real time, therefore you should turn the function on, take your measurements and then turn it off.

```
-> spyHelp
spyHelp          Print this list
spyClkStart [ticksPerSec] Start task activity monitor running
                  at ticksPerSec ticks per second
spyClkStop       Stop collecting data
spyReport        Prints display of task activity
                  statistics
spyStop          Stop collecting data and reports
spy [freq[,ticksPerSec]] Start spyClkStart and do a report
                  every freq seconds
```

ticksPerSec defaults to 100. freq defaults to 5 seconds.

value = 1 = 0x1

sysConfigFileShow

sysConfigFileShow

Display the system configuration file.

The system configuration file is parsed by the MAM application task to configure the routes, the DSP Codecs and parameters, the Call Server ELAN IP address (active ELNK), the SNMP traps, the DSCP bits, and the 802.1Q priority bits. The file is fully parsed on card reset. It is also parsed when the file is transmitted by OTM (or TM) or when it is transferred by EM.

Changes in some sections of the file (the [TlanConfig] section, for example) require a card reboot before taking effect. Sections that can be changed without a reboot include:

```
[snmp]
[routes]
[dsp0]
[tos]
[ElanConfig]
[Loss Plan]
[firmware]
```

Note: The [routes] section is used to create static route entries on the ELAN interface only. The entry includes the destination IP address, and the subnet mask.

```
-> sysConfigFileShow
[keycode]
keycodeId=12345678-12345678-12345678
```

```
[Security]
ElanAccessOnly=0
```

```
[snmp]
rdCommunityName=public
wrCommunityName=private
trapsEnabled=1
trapsub=0.0.0.0
ts1=0.0.0.0,255.255.254.0,
CardIP=47.11.255.170
sysHostName=SigServ
sysLocation=(null)
sysContact=(null)
```

```
[routes]
```

```
[dsp0]
EchoCancel=1
DspEchoTail=128
VadThreshold=-17
IdleNoise=-65
DtmfToneDetect=1
ModemDetect=1
```

FaxDetect=1
FaxRate=14400
FaxPlayOutNomD=100
FaxActiveTimeout=20
FaxPacketSize=30
Codec=1
VxPayload=20
VxPlayOutNomD=40
VxPlayOutMaxD=80
VadEnabled=0
Codec=9
VxPayload=30
VxPlayOutNomD=60
VxPlayOutMaxD=120
VadEnabled=0
Codec=8
VxPayload=1
VxPlayOutNomD=40
VxPlayOutMaxD=80
VadEnabled=0

[tos]
controlPrio=160
voicePrio=184
802.1pqEnabled=0
802.1p=6
802.1q=0
natEnabled=0
natTimeout=90

[ElanConfig]
CallServerIP=47.11.254.185
SurvivalIP=0.0.0.0
SignalPort=15000
BroadcastPort=15001

[TlanConfig]
SignalPort=5000
AudioPort=5200

[GateKeeper]
PrimaryGKIP=47.11.249.176
AlternateGKIP=47.11.215.243

PrimaryNCSIP=47.11.249.176
PrimaryNCSPort=16500
AlternateNCSIP=47.11.215.243
AlternateNCSPort=16500
NCSTimeout=10

[firmware]
serverIP=0.0.0.0
subnetMask=255.255.254.0
fwfileDirPath=download/firmware/
userID=(null)
password=(null)

[ApplicationServer_47.11.255.170]
HostName=LeadeSS
H323ID=SigServ
SW_VtrkTPS=1
SW_GateKeeper=1
SW_SetTPS=1

[SNTP Server]
Mode=active
Interval=256
Port=21999

[SNTP Client]
Mode=passive
Interval=256
Port=21999
ServerIP=0.0.0.0

[OM Thresholds]
PacketLoss=20
Latency=75
Jitter=40
PollingPeriod=20
CallServerReporting=1

value = 0 = 0x0

syslogLevelSet

syslogLevelSet <task>, <level>

Specify the type of detailed information to be printed from the specified task. To change the type of information to be printed from multiple tasks, the command must be entered for each individual task. The parameters are as follows:

<task> is the specified task name. Use the name of the task as printed by the **syslogShow** or **i** command.

<level> specifies what type of information is printed for the task specified by <task>. Associating a task with a particular level causes messages of the specified level, and lower, to be printed for the specified task. By default, the level associated with most tasks is Info. When troubleshooting a problem, setting <level> to Debug will cause more detailed information to be printed for <task>. Values are:

- 0—Emergency, system unusable
- 1—Alert, action must be taken immediately
- 2—Critical, critical conditions
- 3—Error, error conditions
- 4—Warning, warning conditions
- 5—Notice, normal but significant condition
- 6—Info, informational
- 7—Debug, debug level messages

Note: The volume of messages printed by most tasks when <level> is set to Debug dictates that this command be used only under light traffic loads. Use caution when setting <level> to Debug on a busy card.

```
-> syslogLevelSet tMAM, 7
value = 6 = 0x6
```

syslogShow

syslogShow

Display each task and the associated logging level. This command can be issued in the VxWorks shell only.

```

-> syslogShow
      Task      Level
-----
tSysWork      none
tExcTask      none
tLogTask      none
tAioWait      none
tAioIoTask1   none
tAioIoTask0   none
tNetTask      none
tPortmapd     none
tFtpdTask     none
tTftpdTask    none
tSnmpd        none
tWdbTask      none
tRptd         none
tLogin        none
tLLogin       none
tRLogind      none
tTelnetd      none
tPxTimer      none
tbootpd       none
tSntpTask     none
tRDP          Info
tMAM          Info
tOMM          Info
tRPCMGMT      none
tPBX          Info
tELC          Info
tVTM          Info
tVTI          Info
tSET          Info
tCSV          Info
tTPS          Info
tfwBk         Info
tUMS          Info
tUMC          Info
tVTK          Info
tNPM          Info
tHTTPd        none
tHTTPd        none
tHTTPd        none
tHTTPd        none

```



```
tTelnet35      none
tLogin         none
tTelnetc35     none
shell         none
shell         none
tTelnet38      none
tLogin         none
tTelnetc38     none
shell         none
tTelnet41      none
tLogin         none
tTelnetc41     none
shell         none
tShell        none
value = 5 = 0x5
```

sysRIsModuleShow

sysRIsModuleShow

Display the module version.

The following example was executed on a Release 2.0 system.

```
-> sysRIsModuleShow
(null)          sse-2.00.70
value = 33 = 0x21 = '!'
```

sysRIsInfoShow

sysRIsInfoShow

Display the release information.

The following example was executed on a Release 2.0 system.

```
-> sysRIsInfoShow
sse-2.00.70 Monday October 07 2002 13:55:58 EDT
```

Loaded Modules:

```
share.obj      sse-2.00.70
line.obj       sse-2.00.70
```

```
trunk.obj      sse-2.00.70
gk.obj         sse-2.00.70
web.obj        sse-2.00.70
value = 0 = 0x0
```

tcpstatShow

tcpstatShow

Display the TCP protocol statistics.

The following example was executed on a Release 2.0 system.

```
-> tcpstatShow
TCP:
  168892 packets sent
    41402 data packets (2538862 bytes)
    9 data packets (3600 bytes) retransmitted
    127442 ack-only packets (693 delayed)
    0 URG only packet
    0 window probe packet
    0 window update packet
    39 control packets
  138812 packets received
    11929 acks (for 2538993 bytes)
    119 duplicate acks
    0 ack for unsent data
    130142 packets (36022147 bytes) received in-sequence
    1 completely duplicate packet (320 bytes)
    0 packet with some dup. data (0 byte duped)
    64 out-of-order packets (24 bytes)
    0 packet (0 byte) of data after window
    0 window probe
    1 window update packet
    0 packet received after close
    0 discarded for bad checksum
    0 discarded for bad header offset field
    0 discarded because packet too short
  14 connection requests
  60 connection accepts
  74 connections established (including accepts)
  82 connections closed (including 14 drops)
```

```

0 embryonic connection dropped
11881 segments updated rtt (of 11885 attempts)
0 retransmit timeout
    0 connection dropped by rexmit timeout
0 persist timeout
0 keepalive timeout
    0 keepalive probe sent
    0 connection dropped by keepalive
2 pcb cache lookups failed
value = 28 = 0x1c

```

ttyShow

ttyShow

Display the attributes of all TTYs, or a specified TTY.

```

-> ttyShow
FD = 0
Name = /pty/pty02.S
Options: LINE ECHO CRMOD FLOW 7Bit MON ABORT
Installed hooks: none
value = 0 = 0x0

```

udpstatShow

udpstatShow

Display the UDP protocol statistics.

```

-> udpstatShow
UDP:
9960650 total packets
5122321 input packets
4838329 output packets
0 incomplete header
0 bad data length field
0 bad checksum
84139 broadcasts received with no ports
0 full socket
4814139 pcb cache lookups failed
5 pcb hash lookups failed

```

value = 27 = 0x1b

userMemShow

userMemShow

Display the User Memory Allocation Map.

```
-> userMemShow
User Memory Area Info
-----
Start Address = 0x0cc00000
Size in bytes = 0x03400000
Page Size    = 0x1000
```

Allocation Table

```
Offset Size  Key
-----
0x0000 0x0400 0x0442dfe8 __sysProtHeapKey
0x0400 0x00be 0x0442edbc __rptUserMemKey
value = 0 = 0x0
```

vtrkShow

vtrkShow <protocol>

Display a summary of the VTRK configuration for the specified protocol, where:

<protocol> is the specified protocol. Values are:

- ALL—display the summary for both H.323 and SIP protocols
- H323—display the summary for the H.323 protocol
- SIP—display the summary for the SIP protocol

If no <protocol> parameter is provided, the summary of the VTRK configuration for both the H.323 and SIP protocols is printed.

```
oam> vtrkShow ALL
```

```
-----
```

VTRK Summary

```

-----
VTRK status : Active
Protocol    : SIP and H323
D-Channel  : 10
Customer   : 0
Channels Idle : 12
Channels Busy : 0
Channels Mbsy : 0
Channels Pend : 0
Channels Dsbl : 0
Channels Ukwn : 0
Channels Total: 12
Chid ranges : 1 to 15
    
```

VTRK State = Active

VTRK Status = Enabled

oam> vtrkShow SIP

VTRK Summary

```

-----
VTRK status : Active
Protocol    : SIP
D-Channel  : 10
Customer   : 0
Channels Idle : 6
Channels Busy : 0
Channels Mbsy : 0
Channels Pend : 0
Channels Dsbl : 0
Channels Ukwn : 0
Channels Total: 6
Chid ranges : 10 to 15
    
```

-
**IND TN DCH PROTOCOL CHID CUST ROUTE MEMB ICOG VoIP
ESN5 PRFX SAT NCOS STATUS**

-
6 124-0-00-00 010 MCDN->EST 010 00 005 001 IO SIP NO --- NO
00 CS IDLE
7 124-0-00-01 010 MCDN->EST 011 00 005 002 IO SIP NO --- NO
00 CS IDLE
8 124-0-00-02 010 MCDN->EST 012 00 005 003 IO SIP NO --- NO
00 CS IDLE
9 124-0-00-03 010 MCDN->EST 013 00 005 004 IO SIP NO --- NO
00 CS IDLE
10 124-0-00-04 010 MCDN->EST 014 00 005 005 IO SIP NO --- NO
00 CS IDLE
11 124-0-00-05 010 MCDN->EST 015 00 005 006 IO SIP NO --- NO
00 CS IDLE

-
VTRK State = Active

VTRK Status = Enabled

vtrkShow (range of channels)

vtrkShow <protocol> <start> <quantity>

Display a summary of the VTRK configuration of the specified protocol for a range of channels, where:

<protocol> is the specified protocol. Values are:

ALL—display the summary for both H.323 and SIP protocols

H323—display the summary for the H.323 protocol

SIP—display the summary for the SIP protocol.

<start> is the first channel in the range.

<quantity> is the number of channels to be displayed beginning with channel <start>.

If the <protocol> parameter is not provided, the summary is printed for both the H.323 and SIP protocols.

```
oam> vtrkShow SIP 10 3
```

```
-----  
VTRK Summary
```

```
-----  
VTRK status : Active  
Protocol : SIP  
D-Channel : 10  
Customer : 0  
Channels Idle : 6  
Channels Busy : 0  
Channels Mbsy : 0  
Channels Pend : 0  
Channels Dsbl : 0  
Channels Ukwn : 0  
Channels Total: 6  
Chid ranges : 10 to 15
```

```

-----
-
IND  TN    DCH PROTOCOL CHID  CUST ROUTE MEMB  ICOG VoIP
ESN5 PRFX SAT  NCOS STATUS
-----
-
  6 124-0-00-00 010 MCDN->EST 010 00 005 001 IO SIP NO --- NO
00 CS IDLE
  7 124-0-00-01 010 MCDN->EST 011 00 005 002 IO SIP NO --- NO
00 CS IDLE
  8 124-0-00-02 010 MCDN->EST 012 00 005 003 IO SIP NO --- NO
00 CS IDLE
-----
-

```

VTRK State = Active

VTRK Status = Enabled

Signaling Server tasks

Enter the command `i` in the VxWorks shell to display the list of tasks running on the Signaling Server.

-> i

NAME	ENTRY	TID	PRI	STATUS	PC	SP	ERRNO	DELAY
tExcTask	_excTask	cb68c00	0	PEND	41534e6	cb68b70	3006b	0
tLogTask	_logTask	cb662fc	0	PEND	41534e6	cb66268	0	0
tSysWork	408f7e0	cb73e08	1	DELAY	40a9cbe	cb73dbc	3d0001	33
tShell	_shell	8b1fd3c	1	READY	40d48b0	8b1f9fc	0	0
tWdbTask	40f18e0	bb59dd0	3	PEND	40a83e8	bb59d28	0	0
tPxTimer	_pxTaskInit	ba24b9c	10	DELAY	40a9cbe	ba24b30	4	3591
tAioIoTask1	_aioIoTask	cb44f30	50	PEND	40a83e8	cb44ed8	c0002	0
tAioIoTask0	_aioIoTask	cb3dd9c	50	PEND	40a83e8	cb3dd44	3d0002	0
tNetTask	_netTask	bd701c4	50	PEND	40a83e8	bd7016c	41	0
tAioWait	_aioWaitTask	cb4c0c4	51	PEND	40a83e8	cb4be08	0	0
tFtpdTask	4022090	bcd230c	55	PEND	40a83e8	bcd2240	0	0
tTftpdTask	_tftpdTask	bccf164	55	PEND	40a83e8	bcecbdc	d0003	0


```

tSntpTask 40d7760 b9ee51c 56 PEND 40a83e8 b9ee34c 0 0
tPortmapd _portmapd bcd3870 100 PEND 40a83e8 bcd34a0 16 0
tLogin _taskEntry__ ba73dd8 100 DELAY 40a9cbe ba73ce4 0 40
tLogin _taskEntry__ ba6812c 100 DELAY 40a9cbe ba680a0 1c0001 40
tRLogind _rlogind ba64250 100 PEND 40a83e8 ba64178 0 0
tTelnetd _telnetd ba638bc 100 PEND 40a83e8 ba637c8 0 0
shell _taskEntry__ b9d6804 100 PEND 40a83e8 b9d6484 3d0001 0
tTelnets43 _telnetTask cbfeab4 100 PEND 40a83e8 cbfe75c d0003 0
tLogin _taskEntry__ 8b41680 100 DELAY 40a9cbe 8b415f4 1c0001 28
tTelnetc43 40917d0 8b259f0 100 READY 40a834f 8b25744 0 0
shell _taskEntry__ 8b24cf0 100 DELAY 40a9cbe 8b24b30 30065 11
tRDP _rudpMgrStar b9ec1ec 120 PEND 40a83e8 b9ebb18 b 0
tPBX _pbxTcpRecvT b9ba55c 120 PEND 40a83e8 b9ba408 0 0
tSnmpd 40e7e20 bccc0cc 150 PEND 40a83e8 bccb748 0 0
tbootpd _cmain b9f1c74 200 PEND 40a83e8 b9f1480 0 0
tMAM _mamMain b9dabec 200 PEND 411cdf8 b9dab14 0 0
tRPCMGMT _start_ss_se b9ca754 200 PEND 40a83e8 b9ca394 3d0004 0
tELC _electTask b9b83c8 200 PEND 40a83e8 b9b7ce4 11 0
tVTM 22fba70 b958b18 200 PEND 411cdf8 b958a38 4 0
tSET 2296db8 b940158 200 PEND 411cdf8 b94007c 4 0
tCSV _csvTask b936014 200 PEND 411cdf8 b935f30 4 0
tTPS _tpsTask b92f87c 200 PEND 40a83e8 b92f010 380003 0
tUMS _umsServerSt b90ad18 200 PEND 411cdf8 b90aba8 4 0
tUMC _umsClientSt b0e726c 200 PEND 411cdf8 b0e718c 3d0004 0
tVTK _vtrkTask b0db41c 200 PEND 411cdf8 b0db340 3d0004 0
tNPM _npmMain b0c90c4 200 PEND 40a83e8 b0c8c5c 39 0
tGKNPM _gkNpmMain 996a584 200 PEND 40a83e8 996a120 39 0
tGKVONPM _gkVoMain 9432b64 200 PEND 40a83e8 943295c 3d0004
0
tGKDBM _gkDbmMaintM 9997b60 201 PEND 411cdf8 9997a7c 4 0
tOMM _ommMain b9d16f4 250 PEND 411cdf8 b9d1620 380003 0
tVTI _ytiTask b94675c 250 PEND 411cdf8 b946688 3d0004 0
tGKOMM _gkOmmMain 9971170 250 PEND 411cdf8 997109c 4 0
tHTTpd 26636d8 8d25130 250 PEND+T 40a83e8 8d24d2c 3d0004 14669
tHTTpd 26636d8 8cd4f9c 250 PEND+T 40a83e8 8cd4b98 3d0004 14669
tHTTpd 26636d8 8c84e08 250 PEND+T 40a83e8 8c84a04 3d0004 14668
tHTTpd 26636d8 8c34c74 250 PEND+T 40a83e8 8c34870 3d0004 14668
tRptd 407f588 ba87d74 255 PEND 40a83e8 ba87a0c c0002 0
tfwBk _umsServerFw b90c61c 255 PEND 411cdf8 b90c544 1c0001 0
value = 0 = 0x0

```

Signaling Server Quality of Service (QoS)

This section provides an overview of QoS monitoring. QoS monitoring applies to both the Signaling Server and the VGMC cards.

QoS monitoring

- Monitors packet loss, jitter, and latency of the registered IP telephones.
- Monitors packet loss and Jitter for Voice Gateways (VGW).
- Raises an alarm when the threshold for packet loss, jitter, or latency are exceeded. Alarm thresholds are set in the **config.ini** file as follows:

[OM_Thresholds]

Jitter = 40

Latency = 75

Packetloss = 20

PollingPeriod = 20

CallServerReporting = 1

QoS alarm thresholds

The QoS alarm thresholds can be modified through Element Manager or by using a text editor.

To modify the thresholds in Element Manager, select **Node Summary > IP Telephony: Node ID XXXX**. Select the **OM Thresholds** section and set the desired threshold values. A system reboot is not required for the new values to take effect.

The QoS alarm thresholds are:

- 1 **CallServerReporting**—activate/deactivate the QoS alarm reporting to the call server. Values are:
 - 0—off (Default),
 - 1—on.
- 2 **PollingPeriod**—the QoS polling interval for IP telephones and DSP channels. Values range from 20 to 600 seconds.

- 3 **PacketLoss**—the violation threshold for packet loss. Values range from 0 - 500 in units of 1/10th of a percent.
- 4 **Latency**—the violation threshold for delay. Values range from 0 to 2000 milliseconds, where 0 means no monitoring against this threshold.
- 5 **Jitter**—the violation threshold for variation in latency. Values range from 0 to 2000 milliseconds, where 0 means no monitoring against this threshold.

Note: A value of 0 in any of these thresholds indicates that no monitoring or reporting against that threshold will take place.

QoS operation

QoS uses a set polling period to poll telephones and voice gateways for statistics. If a violation of a threshold level occurs, the corresponding ITG alarm is generated. The ITG alarms are as follows:

ITG4028—packet loss threshold violation,

ITG4043—latency threshold violation,

ITG4044—jitter threshold violation.

After 5 successive polls with no errors, the following alarm clears are generated:

ITG5028—packet loss violation alarm cleared,

ITG5043—latency violation alarm cleared,

ITG5044—jitter violation alarm cleared.

QoS settings and troubleshooting

OmmQoSPollingConfigShow

Display the current QoS settings. This command is executed from the vxshell.

-> **ommQoS**PollingConfigShow

=== OMM QoS Polling Configurations ===

```
setCallServerReporting = ON
setQoS polling = ON
setPollingPeriod = 20 secs
dspCallServerReporting = ON
dspQoS polling = ON
dspPollingPeriod = 20 secs
packetLossThreshold = 20
latencyThreshold = 75
jitterThreshold = 40
monitorPacketLoss = ON
monitorLatency = ON
monitorJitter = ON
QoSViolationMsgQLength = 100
processQoSViolationPeriod = 30 secs

value = 38 = 0x26 = '&'
->
```

Voice Gateway troubleshooting

Enhanced trace tools

There are three trace tools available for troubleshooting on VGMC cards:

- **UNISstim trace tool**
- **VGW trace tool**
- **DCH trace tool**

UNISstim trace tool

The **H323CallTrace** commands apply to both the Signaling Server and the VGMC cards. For more information, see “UNISstim trace tool commands” on [page 51](#).

VGW trace tool

The VGW trace tool consists of a set of commands used for troubleshooting on VGMC cards only. The command are executed in the VxWorks shell.

- **vgwTraceHelp**

Display the help information for vgwTrace.

- **vgwTraceSetOutput <output_dest>**

Direct the vgwTrace output to the specified output destination, where:

<output_dest> is the output destination. Values are:

- 1—TTY
- 2—RPTLOG

- **vgwTraceOff <channelNum>, <trace_tool>**

Deactivate the specified VGW trace tool for the specified channel, where:

<channelNum> is the channel number. Values range from 0 - maximum channel number;

<trace_tool> is the type of VGW trace. Values are:

- 0—All VGW message tracing
- 1—A07 message tracing
- 2—Audio message tracing
- 4—Registration message tracing

Note: To disable more than one type of trace, sum the values of <trace_tool> for the desired trace types and enter the sum as the parameter. For example, to deactivate both A07 and Audio message tracing, enter “3” as the parameter.

- **vgwTraceAllOff**

Deactivate all VGW traces for all channels.

- **vgwTraceOn <channelNum>, <trace_tool>, <output_dest>**

Activate the specified VGW trace tool for one channel and direct the trace output to <trace output> where:

<channelNum> is the channel number. Values range from 0 - maximum channel number

<trace_tool> is the type of VGW trace. Values are:

0—All VGW message tracing

1—A07 message tracing

2—Audio message tracing

4—Registration message tracing

Note: To enable more than one type of trace, sum the values of <trace_tool> for the desired trace types and enter the sum as the parameter. For example, to activate both A07 and Audio message tracing, enter “3” as the parameter.

<output_dest> is the output destination. Values are:

1—TTY

2—RPTLOG

- **itgA07TraceHelp**

Display the A07 specific help menu.

- **itgA07TraceSettings**

Display the current A07 trace settings.

- **itgA07TraceSetOutput <output_dest>**

Direct the itgA07 trace output to the specified output destination, where:

<output_dest> is the output destination. Values are:

1—TTY

2—SYSLOG

- **itgA07TraceOff <channelNum>**

Deactivate the A07 trace for the specified channel, where:
<channelNum> is the channel number. Values range from 0 - maximum channel number.
- **itgA07TraceAllOff**

Deactivate the A07 trace for all channels.
- **itgA07TraceOn <channelNum>,<output_dest>**

Activate an A07 trace on one channel and direct the trace output to the specified output destination, where:
<channelNum> is the channel number. Values range from 0 - maximum channel number;
<output_dest> is the output destination. Values are:
1—TTY
2—SYSLOG
- **vgwAudioTraceHelp**

Display the help menu for audio trace.
- **vgwAudioTraceSettings**

Displays the current trace settings.
- **vgwAudioTraceSetOutput <output_dest>**

Direct the itgA07 trace output to the specified output destination, where:
<output_dest> is the output destination. Values are:
1—TTY
2—SYSLOG
- **vgwAudioTraceOff <channelNum>**

Deactivate audio trace for the specified channel, where:
<channelNum> is the channel number. Values range from 0 - maximum channel number.

- **vgwAudioTraceAllOff**
Deactivate audio trace for all channels.
- **vgwAudioTraceOn <channelNum>,<output_dest>**
Activate an audio trace for one channel and direct the trace output to the specified output destination, where:

<channelNum> is the channel number. Values range from 0 - maximum channel number;

<output_dest> is the output destination. Values are:
1—TTY
2—SYSLOG
- **vgwRegisterTraceHelp**
Display the registration help menu.
- **vgwRegistrationTraceSettings**
Display the current registration trace settings.
- **vgwRegistrationTraceSetOutput <output_dest>**
Direct the registration trace output to the specified output destination, where:

<output_dest> is the output destination. Values are:
1—TTY
2—SYSLOG
- **vgwRegistrationTraceOff <channelNum>**
Deactivate the registration trace for one channel, where:

<channelNum> is the channel number. Values range from 0 - maximum channel number
- **vgwRegisterTraceAllOff**
Deactivate the registration trace for all channels.

- **vgwRegistrationTraceOn <channelNum>, <output_dest>**

Activate the registration trace for one channel and direct the trace output to the specified output destination, where:

<channelNum> is the channel number. Values range from 0 - maximum channel number;

<output_dest> is the output destination. Values are:

1—TTY

2—SYSLOG

DCH trace tool

The DCH trace or diagnostic commands apply to both the Signaling Server and the VGMC cards. For more information, see “DCH diagnostic tool” on [page 29](#).

IP Peer Networking and Gatekeeper

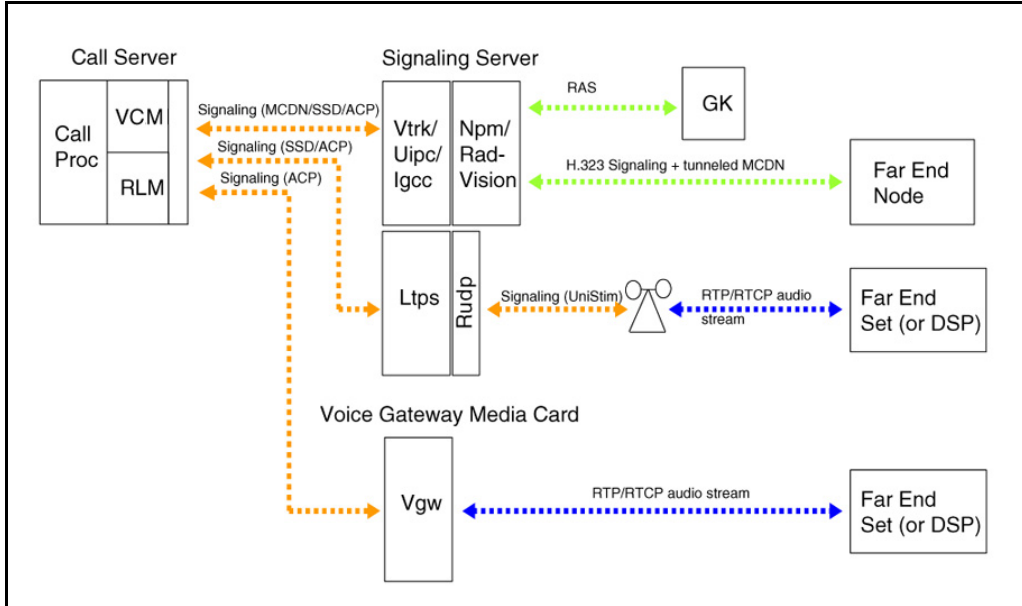
This chapter provides an in-depth description of the tools and utilities available to aid in the analysis and diagnosis of IP Peer Networking problems. The primary software components of IP Peer Networking are covered as follows:

- “Network Routing Server” on [page 148](#)
- “SIP Redirect Server” on [page 148](#)
- “Gatekeeper” on [page 155](#)
- “Virtual trunk on the Call Server” on [page 168](#)
- “Virtual Trunk on the Gateway” on [page 178](#).

Software components

Figure 3: “Software components and message paths” on [page 148](#) illustrates how the primary software components on the Call Server, the Signaling Server, and the Voice Gateway Media Card communicate with one another.

Figure 3
Software components and message paths



Network Routing Server

The SIP Redirect Server and the H.323 Gatekeeper both reside on the Network Routing Server (NRS).

SIP Redirect Server

The SIP Redirect Server is a software component of the NRS. It provides address resolution for CS 1000 and Succession 2000 call servers. The SIP Redirect Server provides dynamic registration and SIP support.

The Redirect Server requires one of the following hardware platforms:

- CS 1000 Release 4.0 on an ISP1100 (Signaling Server) running VxWorks
- Succession 2000 Release 08 on a HP server (NEBS compliant) running Nortel NCGL.

The SIP Redirect/Proxy Server consists of four major software components:

- SIP Proxy/Redirect Server logic (including the oSIP protocol stack)
- Network Routing Service (NRS) database (based on the Solid Database Engine)
- Web Server (including CGI-based server side logic)
- Web Client (including Java Script-based client side logic)

The NRS database is preconfigured with the routing information through a web-based configuration tool. The configuration tool server side CGI interacts with the database using an ODBC interface.

The SIP Proxy/Redirect Server logic routes (directly or indirectly) SIP requests (typically representing new call attempts) to the proper destination. The SIP engine queries the NRS database for the signaling transport address of the appropriate SIP termination endpoint. The query is based on the domain, the phone context and the actual DN digits contained in the SIP request URI being processed. The database query is performed using the ODBC interface.

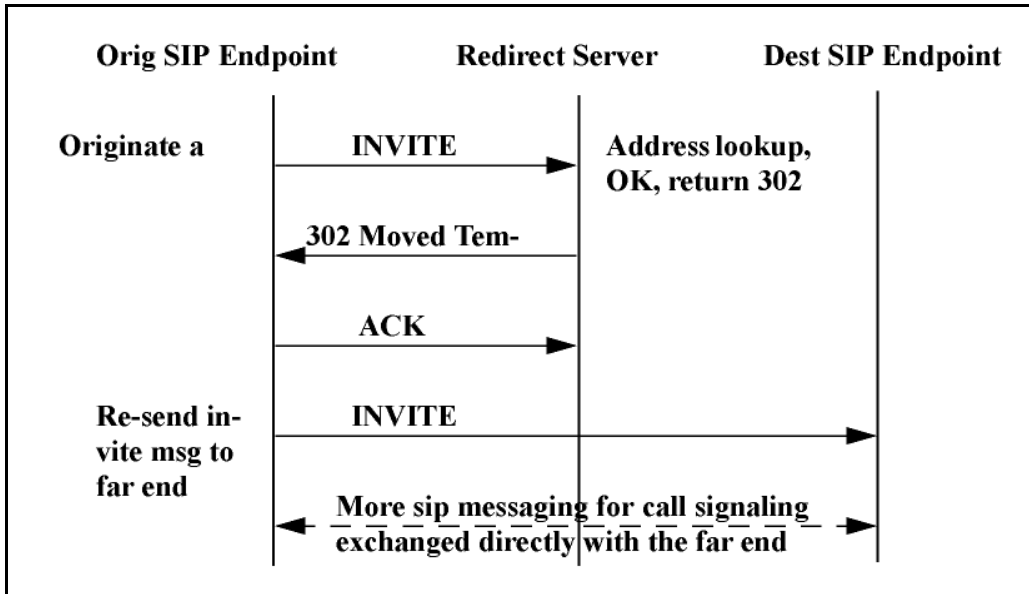
The SIP Proxy/Redirect Server supports both UDP and TCP signaling transports.

Redirect Server troubleshooting

Basic message flow for a successful call

Figure 4: “Message flow—successful call” on [page 150](#) illustrates the message flow for a successful call.

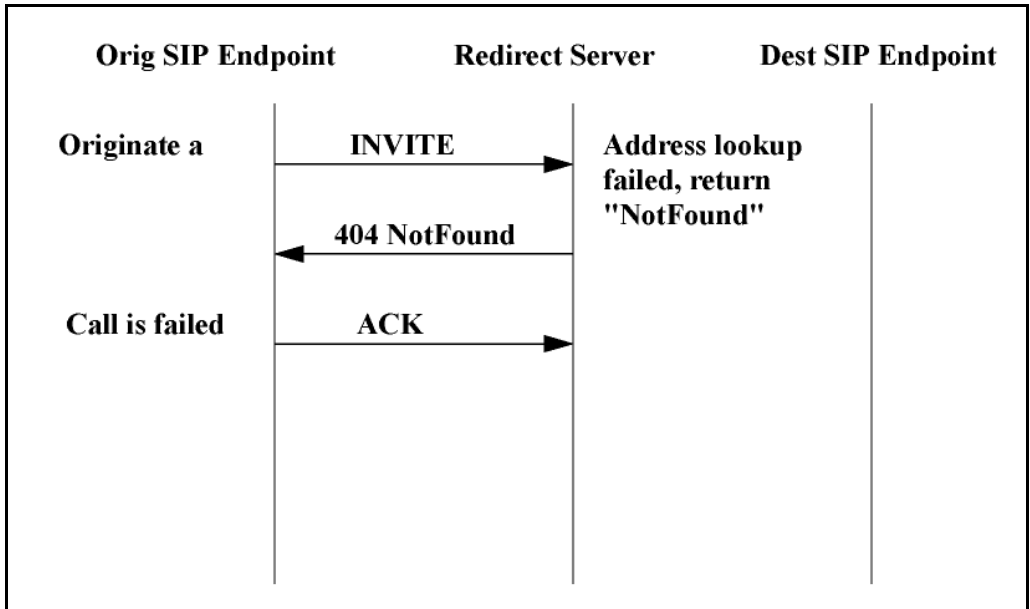
Figure 4
Message flow—successful call



Basic message flow for a failed Redirect Server lookup

Figure 5: “Message flow—failed Redirect Server lookup” on [page 151](#) illustrates the message flow that occurs when a Redirect Server lookup fails.

Figure 5
Message flow—failed Redirect Server lookup



Redirect Server shell commands

To determine whether the appropriate routing information exists within the database, issue the following command:

```

-> nrsSIPQuery "myServiceProvider.com", "myCdpDomain.myCompany.com", "4", "47.11.242.212",0,0
Entry #0: RouteCost=1 FqdnType=0 sHostFqdn=47.11.242.211 Transport=1
Port=5060 L1Name=myCompany.com L1Auth=0 L1Pwd=
GWEName=sipGW1 GWEAuth=2 GWEPwd=
RouteType=REGULAR_ROUTE
Entry #1: No entry found.
value = 0 = 0x0
  
```

Redirect Server SIP message tracing

- **SpMsgTrace = <traceActivate>**

Activate or deactivate single syslog message tracing. Values for <traceActivate> are:

 - 0—deactivate the trace
 - 1—activate the trace
- **spLogSet <syslogLevel>**

Set the logging level for all Redirect Server tasks with one command. For a list of the Redirect Server tasks see “Redirect Server tasks” on [page 153](#). Values for <syslogLevel> are:

 - 6—event
 - 7—debug
- **syslogLevelSet <task>, 7**

Set the logging level for the specified Redirect Server task to Debug. Debug messages generated from the task are stored in the syslog. For a list of the Redirect Server tasks see “Redirect Server tasks” on [page 153](#).

```
-> SpMsgTrace = 1
SpMsgTrace = 0x2c21034: value = 1 = 0x1
->
03/02/04 00:49:19 LOG0006 tSPMsg: 47.11.181.62:5060 -> INVITE 5702;
phone-context=myCdpDomain.myCompany.com@myServiceProvid-
er.com, Callid 643a87-58b8, tick 1501091
03/02/04 00:49:19 LOG0006 tSPMsg: 47.11.181.62:5060 <- 404 Not Found, Cal-
lid 643a87-58b8, tick 1501091
03/02/04 00:49:20 LOG0006 tSPMsg: 47.11.181.62:5060 -> ACK, Callid
643a87-58b8, tick 1501166

03/02/04 00:55:49 LOG0006 tSPMsg: 47.11.181.62:5060 -> INVITE 5702;
phone-context=myCdpDomain.myCompany.com@myServiceProvider.com,
Callid 1949501c-58b8, tick 1524504
```



```
03/02/04 00:55:49 LOG0006 tSPMsg: 47.11.181.62:5060 <- 302 Moved Temporarily, Callid 1949501c-58b8, tick 1524504
03/02/04 00:55:50 LOG0006 tSPMsg: 47.11.181.62:5060 -> ACK, Callid 1949501c-58b8, tick 1524579
```

Redirect Server tasks

The following list provides the Redirect Server tasks:

- **tSPRouter**—the main internal message dispatcher
- **tSPRe**—the endpoint registration service
- **tSPLoc**—the endpoint lookup service
- **tSPMsg**—the SIP message parsing & state machine
- **tSPUDPd**—the SIP udp transport service
- **tSPTCPd**—the SIP tcp transport service

```
-> i
NAME      ENTRY      TID  PRI  STATUS  PC   SP   ERRNO  DELAY
-----
tSPRouter _spProxyRout b27d18c 200 PEND   314b388 b27d01c 1c0001 0
tSPReg    _spRegistrar b27aff8 200 PEND   314b388 b27af0c 1c0001 0
tSPLoc    _spLocation  b278e64 200 PEND   314b388 b278b18 1c0001 0
tSPMsg    _spMsgHandle b276cd0 200 PEND   314b388 b276bd0 1c0001 0
tSPUDPd   _spUdpServer b274b3c 200 PEND   30dc5f0 b274498 1c0001 0
tSPTCPd   _spTcpServer b2729a8 200 PEND   30dc5f0 b272020 1c0001 0
```

OAM level commands

- **nrsDisableServer**
Gracefully disable the NRS server service.
Note: This command will not interrupt any existing calls.
- **nrsForceDisableServer**
Force disable the NRS server service.

- **nrsEnableServer**
Enable the SIP Redirect Server service.
- **nrsSyncForce**
Force a database synchronization with the master database.
- **nrsGWEndpointShow**
Display all the NRS endpoints with the corresponding IP addresses.
- **nrsL0DomainShow**
Display all level 0 regional domains.
- **nrsL1DomainShow**
Display all level 1 regional domains.
- **nrsServiceDomainShow**
Display all service provider domains.
- **nrsWebUserShow**
Display all the web users who are currently logged on.
- **nrsSIPQuery**
Query a SIP routing entry with DN and cost information.
- **nrsGKQuery**
Query a H.323 routing entry with DN and cost information.
- **nrsGWEndpointQuery**
Query a NRS endpoint with IP and protocol information.
- **nrsL0DomainQuery**
Query a level 0 regional domain with E.164 information.
- **nrsL1DomainQuery**
Query a level 1 regional domain.
- **nrsServiceDomainQuery**
Query a service provider domain.

- **nrsDefaultRouteQuery**
Query a NRS default route.
- **nrsCallTrace**
Trace all SIP messages for a specified endpoint.
- **nrsTraceOff**
Disable message tracing for a specified endpoint.
- **nrsTraceOutput**
Set the output destination for the trace output(tty,rpt,...).
- **nrsTraceSettings**
Display the endpoints and destinations currently being traced.
- **nrsTraceHelp**
List the available **nrs** trace commands.

Gatekeeper

The Gatekeeper (GK) is the brain of an H.323 network. It provides registration, admission, and status (RAS) for the H.323 endpoints. The CS 1000 GK-Lite implementation is a direct-route model that addresses RAS messages only. Call signaling in the GK-Lite takes place directly between endpoints.

RAS messages

RAS messages are UDP/IP messages that are sent and received by the Gatekeeper. Table 3: “RAS messages” on [page 155](#) lists the messages, and provides a brief description of each one.

Table 3
RAS messages

GRQ	GK discovery request. This message is sent by an endpoint.
GCF	GK confirm. This message is sent in reply to GRQ.
GRJ	GK reject, This message is sent in reply to GRQ.

Table 3
RAS messages

RRQ	Registration request sent by an endpoint. It includes the H.323 alias, the RAS address, and the CS address of the sending endpoint. The GK supports both light and full registration request (RRQ).
RCF	Registration confirm
RRJ	Registration reject
URQ	Unregistration request
UCF	Unregistration confirm
URJ	Unregistration reject
ARQ	Admission request
ACF	Admission confirm
ARJ	Admission reject
BRQ	Bandwidth request
BCF	Bandwidth confirm
BRJ	Bandwidth reject
DRQ	Disengage request. This message is received when the call is released.
DCF	Disengage confirm
DRJ	Disengage reject
LRQ	Location request, This message is used between GK zones.
LCF	Location confirm
LRJ	Location reject

Software patching

This section outlines the commands available for patching.

To load a patch on the Signaling Server using Element Manager, upload the patch to the **/u/patch/** directory, using the **ftp** command from the command line.

Note: To load a patch on a stand alone Signaling Server, copy the patch to a floppy and, using vxshell, copy the file to the **/u/patch/** directory.

-> copy "/f0/gkpatch.p","/u/patch/gkpatch.p"

To load a specified patch using the pdt shell:

```
pdt> pload
Patch filename? gkpatch.p
Retain patch (y/n)? [y] y
Days patch vulnerable to sysload? [3] <cr>
In-service initialize threshold? [5] <cr>
In-service days to monitor inits? [7] <cr>
Loading patch from "/u/patch/gkpatch.p"
Patch handle is: 0
```

To place a specified patch into service:

```
pdt> pins 0
function at 0x561f10 will be patched to jump to 0xcff9cac
(_gkNpmHandleARQRequest)
Proceed with patch activation (y/n)? [y] y
Patch 0 has been activated successfully.
```

To display the status of a specified patch:

```
pdt> pstat 0
Patch handle: 0*
  Filename: /u/patch/gkpatch.p
  Patch version: 0.21
  Reference number: 1OF1
  Patch is in-service
  In-service date:      03/09/02 15:16:48
  Last out-of-service date: 01/01/70 00:00:00
  Patch is retained
  Patch retain level: RES
```

To take a specified patch out of service:

```
pd> poos 0  
Are you sure (y/n)? [y] y  
Patch 0 has been deactivated successfully.
```

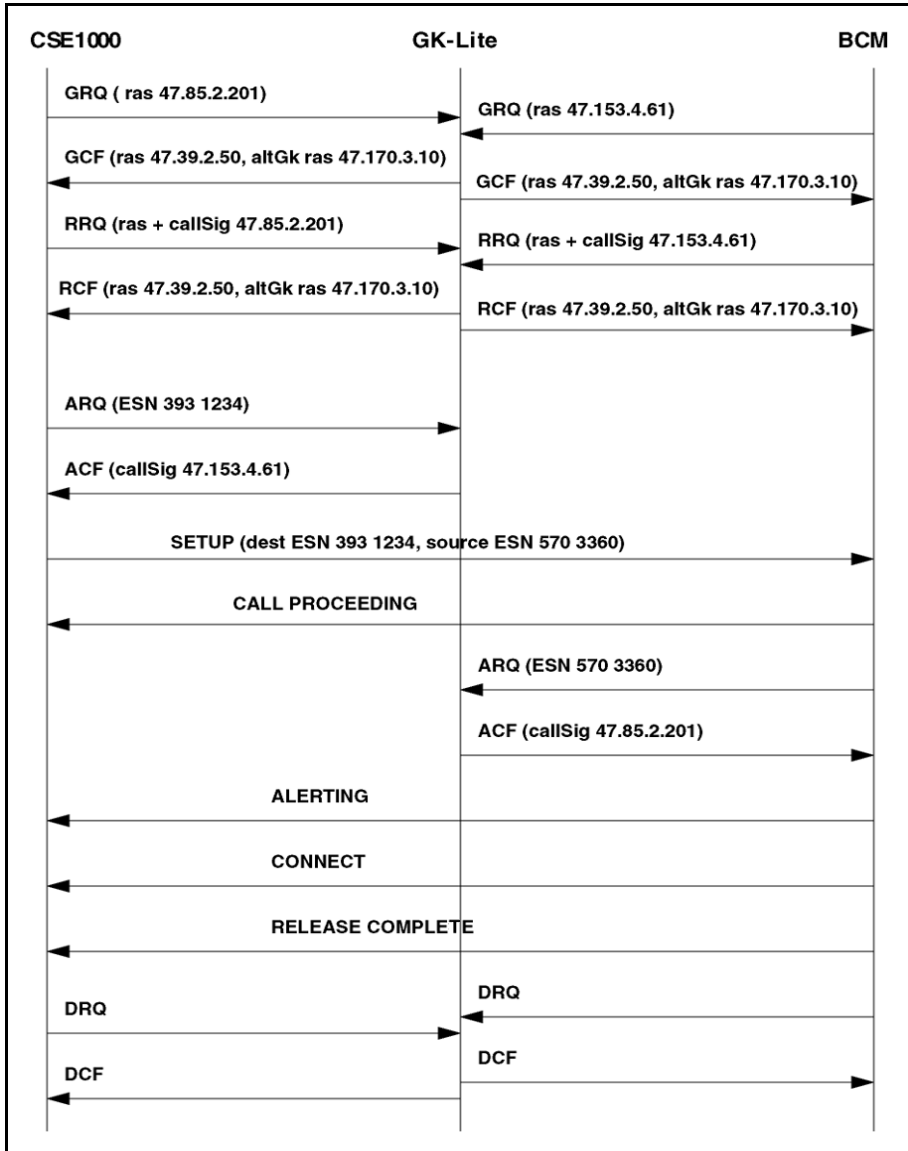
To remove a specified patch:

```
pd> pout 0  
Patch 0 has been removed successfully.
```

Message flow for a complete call

A call flow diagram, including GK discovery and registration, is shown in Figure 6: “Message flow—complete call” on [page 159](#).

Figure 6
Message flow—complete call



Gatekeeper roles

The CS 1000 Gatekeeper can be configured as a primary, alternate or failsafe GK. In order to configure a failsafe GK, there must be both a primary and an alternate GK configured in the network.

Gatekeeper modes

The status of the CS 1000 GK is one of:

- active
- standby
- out of service.

The GK only accepts RAS messages when active. RAS messages will be rejected if the GK is in standby or out of service mode.

Gatekeeper trace

To enable the GK trace for a specified task (for example, tGKNPM):

```
-> syslogLevelSet tGKNPM, 7
```

To disable the GK trace for a specified task (for example, tGKNPM):

```
-> syslogLevelSet tGKNPM,6
```

Gatekeeper detailed stack trace

To enable a GK detailed stack trace from vxshell:

```
->dumpgkpvt = 1
```

To disable a GK detailed stack trace from vxshell:

```
->dumpgkpvt = 0
```

Gatekeeper shell commands

To display the current GK status, role and IP:


```
-> gkShow
Gatekeeper Info
-----
GK Version   : 2.00.53
Role        : Primary GK
Status      : Active
RAS Address  : 47.11.214.132:1719
value = 35 = 0x23 = '#'
```

To display the current status of the endpoints:

```
-> gkEndpointsShow
gkDbmCliEndpointsQuery: DATABASE1 ACTIVE
0x0000000c      SS2 0x005a1018: REGISTERED 0x00607d7c
0x0000000b      abcd 0x005a0e14: UNREGISTERED 0x00607c50
0x0000000a      SS20 0x005a0f6c: UNREGISTERED 0x00607b24
```

To display all GK global variables:

```
-> gkGlobsShow
```

To display the gknpm stack memory information:

```
-> gknpmStackMemoryShow
```

GK-Lite supported alias types

The GK-Lite performs address translations on partyNumber and E.164 aliases only. In general, the partyNumber alias can be one of several subtypes, however, GK-Lite supports only two subtypes: partyNumber.publicNumber and partyNumber.privateNumber. The partyNumber.publicNumber and partyNumber.privateNumber subtypes also have subtypes. The complete list of partyNumber types that GK-Lite supports is:

- publicNumber.unknown
- publicNumber.internationalNumber
- publicNumber.nationalNumber

- publicNumber.networkSpecificNumber
- publicNumber.subscriberNumber
- publicNumber.abbreviatedNumber
- privateNumber.unknown
- privateNumber.level2RegionalNumber
- privateNumber.level1RegionalNumber
- privateNumber.pISNSpecificNumber
- privateNumber.localNumber
- privateNumber.abbreviatedNumber

TON translation in the Gatekeeper trace

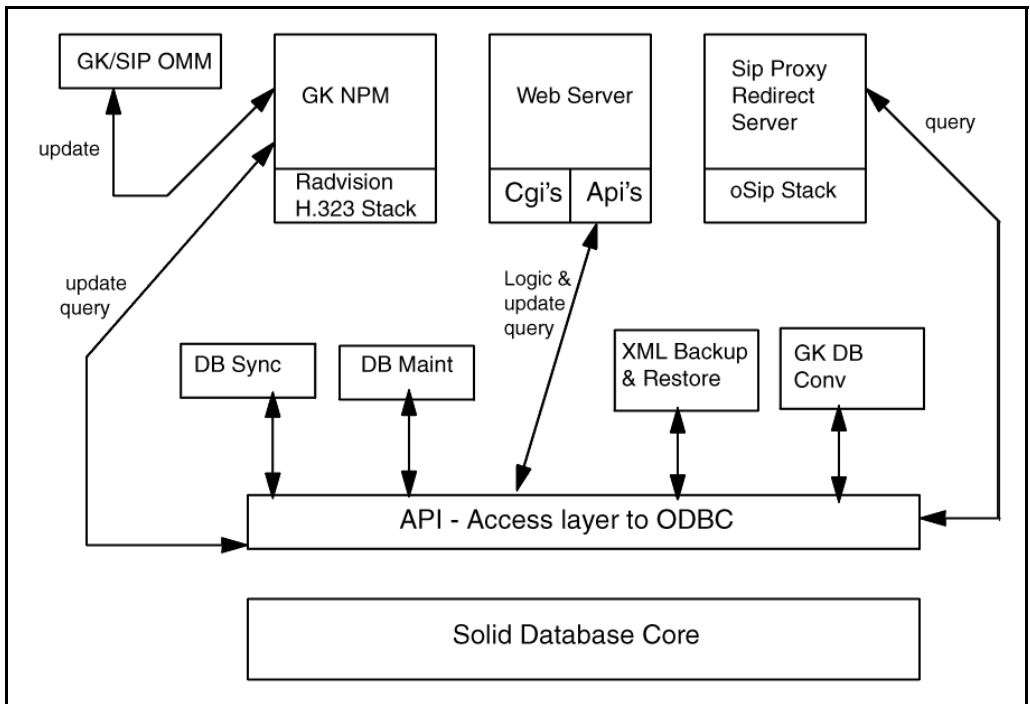
- PARTY_NUMBER_INTERNATIONAL_ALIAS = 1
- PARTY_NUMBER_LEVEL1_ALIAS = 2
- PARTY_NUMBER_PISN_ALIAS = 3
- PARTY_NUMBER_LOCAL_ALIAS = 4
- E164_ALIAS = 5
- H323ID_ALIAS = 6
- URLID_ALIAS = 7
- TRANSPORTID_ALIAS = 8
- EMAILID_ALIAS = 9
- PARTY_NUMBER_PUBLIC_UNKNOWN_ALIAS = 10
- PARTY_NUMBER_NATIONAL_ALIAS = 11
- PARTY_NUMBER_NETWORK_SPECIFIC_ALIAS = 12
- PARTY_NUMBER_SUBSCRIBER_ALIAS = 13
- PARTY_NUMBER_PUBLIC_ABBREVIATED_ALIAS = 14
- PARTY_NUMBER_DATA_ALIAS = 15
- PARTY_NUMBER_TELEX_ALIAS = 16

- PARTY_NUMBER_PRIVATE_UNKNOWN_ALIAS = 17
- PARTY_NUMBER_LEVEL2_ALIAS = 18
- PARTY_NUMBER_PRIVATE_ABBREVIATED_ALIAS = 19
- PARTY_NUMBER_NATIONAL_STANDARD_ALIAS = 20

Gatekeeper software

Figure 7: “Database applications and components” on page 163 shows the relationship between the database application and the database sub-components.

Figure 7
Database applications and components



The primary software tasks of the GK-Lite are:

- **tGKNPM:** Network Protocol Module—encapsulates the RADVision H.323 stack and handles all RAS signaling over the network interface. This task also performs address translations with **tGKDBM**.
- **tHTTPD:** VxWorks HTTP Daemon task—Wind River HTTP daemon task is a component of the Wind Web Server. This task handles the Web configuration interface and any HTML requests over the network interface.
- **tGKDBM:** Database Module—interfaces with the primary and backup database files on the file system. Any changes to the current configuration made via the web interface are passed to **tDataBase**. **tDataBase** updates the database files on disk, synchronizes the primary and backup databases, updates the internal lookup tables, and determines which **tGKNPM** to use.
- **tShell:** VxWorks shell task—Wind River shell task that provides access to support and debug tools. Administrators can telnet to the GK-Lite over the network and access the shell directly.
- **tGKOMM:** Operational Measurement Module—task that provides operational information about the GK-Lite, including traffic measurements, call completion rate (based on successfully handling ARQ), and message monitoring. All application tasks interface with **tGKOMM**.

Database actions

- **Rollback**—cancel changes in the standby database.
- **Cutover**—put the standby database in active mode and the active database in standby mode. No database changes are accepted while in standby mode.
- **Revert**—revert from **Cutover** to the previous database state.
- **Commit**—copy the active database to the standby database.
- **Cutover + Commit**—**Cutover** followed by **Commit** in a single action.

Configuration tips

Alternate Gatekeeper Permanent In Service feature

This feature must be enabled on both the primary and alternate GateKeepers in order to function properly.

CDP LRQ between two Gatekeeper zones

Endpoints in domain (A) of GKPzone1 must be defined in a related CDP domain (B) of GKPzone2 and vice versa.

Note: The aliases only are required.

Test numbering plan

This utility emulates an ARQ received by the Gatekeeper. The required input fields are the number, the type of number, and the originating endpoint. The output is a list of the endpoints that match the query.

National and international numbers

All numbers must be fully qualified E.164 numbers.

- International type—country code followed by city code (for example, 1416 where 1 is the country code and 416 is the city code)
- National type—city code only (for example, . 416 where 416 is the city code)

Use the **CTYP** prompt in the DMI on the CS to assign a type to the number (INTL, NPA).

Admission reject

An endpoint receives an ARJ (Admission Reject) under one of the following conditions:

- Far end endpoint not registered.
- Number not configured, configured with wrong type, or configured with wrong CDP domain.

GKP-GKA polling trace

NSM between the 2 Gatekeepers provides this function.

- To enable the polling trace
dbgnsm = 1
- To disable the polling trace
dbgnsm = 0

Passwords

The GateKeeper application uses 3 passwords:

- Admin
- Guest
- FTP

Manual reset of the passwords is performed from vxshell. Enter the following sequence of commands:

-> **cd “/u/gk/database”**

-> **rm “data.dat”**

The **data.dat** file contains the 3 passwords in an encrypted format.

-> **reboot -1**

The passwords are reset to default values.

Gatekeeper database limits

The Gatekeeper database limits are:

- Maximum number of CDP domain—1000
- Maximum number of endpoints—2000
- Maximum number of numbering plan entries—10000

- Maximum number of default routes—1000
- Maximum number of network zones—20

To check the Gatekeeper limits from the vxshell, enter the command:

```
-> gkDbmCliLimitsShow
```

```
-> gkDbmCliLimitsShow
```

```
---- Gatekeeper Limits ----  
CDP DOMAINS: 1000  
ENDPOINTS: 2000  
NUMBERING ENTRIES: 10000  
DEFAULT ROUTES: 1000  
NETWORK ZONES: 20  
value = 0 = 0x0
```

To change the Gatekeeper limits using vxshell, enter the command:

```
-> gkDbmCliLimitsSet #CDPDomains, #Endpts, #Entries, #Deflt Rtes
```

Note: The number of Gatekeeper network zones cannot be changed.

```
-> gkDbmCliLimitsSet 1000, 2200, 12000, 1000
```

```
Please, reboot the system for the changes to take effect...
```

```
value = 0 = 0x0
```

```
->
```

```
-> reboot -1
```

```
->
```

```
->
```

```
-> gkDbmCliLimitsShow
```

```
---- Gatekeeper Limits ----  
CDP DOMAINS: 1000  
ENDPOINTS: 2200  
NUMBERING ENTRIES: 12000  
DEFAULT ROUTES: 1000  
NETWORK ZONES: 20
```

To reset the Gatekeeper limits to the default values enter the **gkDbmCliLimitsReset** command in vxshell.

```
-> gkDbmCliLimitsReset
Please, reboot the system for the changes to take effect...
value = 0 = 0x0
->
-> reboot -1
```

Virtual trunk on the Call Server

This section outlines how to troubleshoot a failed virtual trunk call. Perform the actions in each section, in sequence, to troubleshoot the call.

srvShow

Display all the Signaling Servers and Media Cards connected to the Call Server, along with their attributes.

This command helps you to determine if the correct Signaling Server is connecting to the Call Server. The output of the command consists of:

- **nodename**—the node ID of the server. It must be the same node id as was configured in overlay 16.
- **hostname**—the host name of the server.
- **leaderFlag**—identifies a server as a leader if the value is 1. There is only one leader per node, The primary Signaling Server must be a leader.
- **serverType**—identifies the server type. Values are:
 - 0—ITG486
 - 1—ITG Pentium
 - 2—ITG SA
 - 3—Signaling Server

- **appBitMap**—identifies the applications that are active on the server.
Values are:

- 4—VTRK is active
- 1—PL is active
- 5—both VTRK and IPL are active

Note: In order for virtual trunk to function, the value of appBitMap must be 4 or 5.

For reference, the bit map of appBitMap is:

- bit 0—IPL
- bit 1—VGW
- bit 2—VTRK
- bit 3—GK
- bit 4—WEB_SERVER

- **IPAddr**—the ELAN IP address of the server.
- **connectID**—the connection ID of the server.

Note: If the Signaling Server registration is unsuccessful, review the message display on both the Call Server side and the Signaling Server side. This problem occurs when a site has an ITGL card configured as a leader and a primary Signaling Server (also a leader) attempts to register. The Call Server only allows one leader per node, therefore, if an ITGL card is already configured as a leader, the Call Server will not accept the registration from the primary Signaling Server, which is also a leader.

The following example illustrates the output of the **srvShow** command.

```

pdt> srvShow
nodename hostname leaderFlag serverType appBitMap IPAddr connectID
8517 vxTarget 0 1 0x1 47.11.217.108 0x200a2048
8517 chanss 1 3 0x5 47.11.217.105 0x200a2128

```

D-channel state

If there is an active Signaling Server connected to the Call Server and the virtual trunk is still not functioning, check the virtual D-channel status in overlay 96 as follows:

LD 96
STAT DCH <dch#>

The display should be:

DCH 063 : OPER EST ACTV AUTO

Use the **DCHstatus** command to check the D-channel status on the Signaling Server side also.

Virtual trunk state

If the virtual trunk is still not functioning, use the **STRM** command in overlay 32 to check the status of the virtual trunk.

LD 32
STRM <cust#> <route#> <start_member> <no. of members>

The display should be as follows:

66 03 IDLE REGISTERED
66 04 IDLE REGISTERED

Note: Use the **vtrkShow** command to check the virtual trunk status on the Signaling Server side. If the virtual trunk is not in CS idle state on the Signaling Server side, disable and re-enable the D-channel on the Signaling Server side to clear the problem.

ISDN message monitor

If all of the virtual trunk components are in working status and the virtual trunk still does not function, enable the ISDN message monitor, make a virtual trunk call and observe the messages.

If no ISDN messages are displayed and the virtual trunk remains in the idle state, the problem is probably a traditional telephony problem.

If the virtual trunk sends out an ISDN message, but the call is released before being established, the Gatekeeper may be rejecting the call, or the far-end may have a problem. To determine the cause of the problem, activate the NPM message monitor on the Signaling Server side.

```
LD 96
ENL MSGO <dch#>
ENL MSGI <dch#>
```

rlmShow

rlmShow <> or <0x<TN>>

Display the resources associated with all TNs (**rlmShow**) or with a specific TN (**rlmShow 0x<TN>**).

In order for a virtual trunk to function, the virtual trunk information must be stored in the RLM table. If the information is not in the table, use the **vtrkNodeShow** command to verify that the trunk is lost.

The following example displays the resource for a specific virtual trunk tn.

Note: The codec marked with an asterisk is the codec in use.

```
pdt> rlmShow 0x6847
TN                HWID                STATUS HOSTIP  TERMIP  PORT  CAP
-----
0x6847 000000000000047.11.217.105:6847 REG 0.0.0.0 47.11.217.114 0x1450
0x00000000
-----
codec bdwth(k) codecCaps desc
-----
* 5      0 0x00000000 0
4        0 0x00000000 0
1        0 0x00000000 0
8        0 0x00000000 0
3        0 0x00000000 0
2        0 0x00000000 0
```

The following example displays the resources for a i2004 telephone.

```

pdt> rlmShow 0x6009
TN                HWID                STATUS HOSTIP    TERMIP    PORT
-----
0x6009 00000000000001800603876c78b6600 REG 47.11.217.105
47.11.217.113 0x1450
CAP
-----
0x00000000
-----
codec bdwth(k) codecCaps desc
-----
5    169  0x00000000 1
8    169  0x00000000 1
16   40   0x00000000 1
20   24   0x00000001 1
24   54   0x00000000 1
    
```

Table 4: “Codec List” on [page 172](#) provides the list of codecs for reference.

**Table 4
Codec List**

Codec Number	Codec Name	Vad	Frames per Packet	Milliseconds per Packet	Config.ini value
0	G.711 Clear Channel	off	30	30	9
1	G.711 Ulaw	off	10	10	1
2	G.711 Alaw	off	10	10	0
3	G.711 Alaw	off	20	20	0
4	G.711 Ulaw	off	20	20	1
5	G.711 Ulaw	off	30	30	1
6	G.711 Ulaw	on	20	20	1

Table 4
Codec List

Codec Number	Codec Name	Vad	Frames per Packet	Milliseconds per Packet	Config.ini value
7	G.711 Ulaw	on	30	30	1
8	G.711 Alaw	off	30	30	0
9	G.711 Alaw	on	10	10	0
10	G.711 Alaw	on	20	20	0
11	G.711 Alaw	on	30	30	0
12	G.711 Ulaw	on	10	10	1
13	G.729A	off	2	20	5
14	G.729A	off	3	30	5
15	G.729A	off	4	40	5
16	G.729A	off	5	50	5
17	G.729AB	on	2	20	7
18	G.729AB	on	3	30	7
19	G.729AB	on	4	40	7
20	G.729AB	on	5	50	7
21	G.729A	off	1	10	5
22	G.729AB	on	1	10	7
23	G.723-6	off	1	30	3
24	G.723-5	off	1	30	2
25	G.723-6	off	2	60	3
26	G.723-5	off	2	60	2
27	G.723-6	off	3	90	3

Table 4
Codec List

Codec Number	Codec Name	Vad	Frames per Packet	Milliseconds per Packet	Config.ini value
28	G.723-5	off	3	90	2
29	T.38 Fax	n/a			8

Vtrk registration

vtrkNodeShow

Display the number of virtual trunks in the VTRK registration, VTRK deregistration and offline pending lists.

```
pd> vtrkNodeShow
vtrkNodeShow --- There are request pending in vtrkRegister == 0
vtrkNodeShow --- There are request pending in vtrkRemove == 0
vtrkNodeShow --- There are request pending in vtrkOffline == 0
```

If the values displayed are non-zero for more than 1 minute, use **strm** in overlay 32 on the Call Server, and **vtrkShow** on the Signaling Server to check for lost virtual trunks. If there are lost virtual trunks, the Call Server must be rebooted to reclaim them.

Note: Virtual trunks may be lost when switching between the primary and follower Signaling Servers.

vcmShow

vcmShow <0x<TN>>

Display the call state of a specified virtual trunk, where:

<0x<TN>> is the tn of the virtual trunk.

The following section illustrates the call states that occur when making a call over a virtual trunk. The description of the other fields provided in the examples are beyond the scope of this document.

Virtual trunk call states

- **Idle**

The call state is “idle” if the virtual trunk (TN 0x6847 in the following example) is idle.

pd> vcmShow 0x6847

index	dir	TN	State	Pending			Codec			1co			
				Near	Far	Req	Dsc	Red	list	m/s	dec	tsW0	tsW1
realTn													
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
	Rx	0x0000	IDLE	0	0	0	0	0	0	0	0	0	0x0000
0x6847													
	Tx	0x0000	IDLE	0	0	0	0	0	0	0	0	0	0x0000

- **Ringing**

When a call is dialed from a TN (0x6009 in the following example) over the virtual trunk, the call state for both the receive and transmit channels are “near end connect request”.

pd> vcmShow 0x6847

index	dir	TN	State	Pending			Codec			1co			
				Near	Far	Req	Dsc	Red	list	m/s	dec	tsW0	tsW1
realTn													
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
	Rx	0x6009	NEAR_CON_REQ	1	0	1	0	0	0	0	0101	0	0
0x0000													
0x6847													
	Tx	0x6009	NEAR_CON_REQ	1	0	1	0	0	0	0	0101	0	0
0x0000													
	Rx	0x6847	IDLE	0	0	0	0	0	0	0	1010	a0a0	0x0009
0x6009													
	Tx	0x6847	IDLE	0	0	0	0	0	0	0	1010	a0a0	0x0009

- **Connected**

The call state is “connect” when the far-end answers.

index tsW1	dir realTn	TN	State	Pending			Req	Dsc	Codec Red	1co list	m/s	dec	tsW0	
				Near	Far	Req								
	Rx	0x6009	CONNECT	1	1	0	0	0	1	2	1	1010	0	0x0000
0x6847	Tx	0x6009	CONNECT	1	1	0	0	0	1	2	1	1010	0	0x0000
	Rx	0x6847	IDLE	0	0	0	0	0	0	0	0	1010	a0a0	0x0009
0x6009	Tx	0x6847	IDLE	0	0	0	0	0	0	0	0	1010	a0a0	0x0009

Virtual Connection Manager message monitor

The Virtual Connection Manager (VCM) manages the connection states of virtual trunks. The VCM executes such operations as opening and closing audio channels, and sending requests to the NPM on the Signaling Server. The operations are performed based on the current state of the virtual trunks and the event arriving from either call processing or NPM.

The message monitor displays the events and the state changes that occur during a virtual trunk call. The monitor can display the information in different levels of detail.

itgLogSetLevel <level>

Activate the VCM message monitor at the specified level of detail, where:

<level> is the specified level of detail. Values are:

- 1—disable syslog
- 2—fatal
- 3—error
- 4—warning
- 5—information
- 6—event
- 7—debugging information
- 8—function entry
- 9—function exit

```
pd> itgLogSetLevel 5
```

The ELAN link between the Call Server and Signaling Server

Use the commands in this section to check the IP connections between the Call Server and Signaling Server.

tcpLinkShow

Display the status of the TCP connections.

```
pd> tcpLinkShow
```

```
Master tcp socketFd: 28  Total connections 2
```

```
Link Down Time: 17
```

```
Total retry time Write: 1  Read: 1
```

IPAddr	Connection socketFd	SendMsgLen	RecvMsgLen	LinkStatus
47.11.217.105	0x200a2048 30	0	0	Link OK
47.11.217.108	0x200a2128 41	0	0	Link OK

rudpShow

Display the status and traffic statistics for all connections.

```

pdt> rudpShow
+-----+-----+-----+-----+-----+-----+
| Port ID | Src IP |Src Port| FD | Task| Data 1 | Data 2 |
+-----+-----+-----+-----+-----+-----+
|537419656|0x2f0bd96f|15000 |29 |13312|0      |0      |
+-----+-----+-----+-----+-----+-----+
|Connect ID| Dst IP |Dst Port| Status  | Msg rcv | Msg sent | Retries|
+-----+-----+-----+-----+-----+-----+
|537534760|0x2f0bd96c|15000 |ESTABLISHED <->|1      |12918  |115  |
|537534536|0x2f0bd969|15000 |ESTABLISHED <->|1      |11489  |126  |
+-----+-----+-----+-----+-----+-----+

```

Virtual Trunk on the Gateway

This section provides tools available for use with a virtual trunk on the gateway.

vtrkShow

vtrkShow < ALL or SIP or H323>

Display the status of all virtual trunks. When a virtual trunk is idle (not in use), its status is CS IDLE. If a trunk is not in use and the status is not CS IDLE, disable and re-enable the D-channel to clear the problem.

```

-> vtrkShow ALL
INDEX TN DCH PROTOCOL CHID CUST R MEM VoIP ESN5 PREFX SAT
NCOS STATUS
0 063-00 023 MCDN->EST 001 00 015 001 H.323 NO --- NO 00
CS IDLE
1 063-01 023 MCDN->EST 002 00 015 002 H.323 NO --- NO 00
CS IDLE
2 063-02 023 MCDN->EST 003 00 015 003 H.323 NO --- NO 00
CS IDLE

```

```

3 063-03 023 MCDN->EST 004 00 015 004 H.323 NO --- NO 00
CS IDLE
4 063-04 023 MCDN->EST 005 00 015 005 H.323 NO --- NO 00
CS IDLE
5 063-05 023 MCDN->EST 006 00 015 006 H.323 NO --- NO 00
CS IDLE
6 063-06 023 MCDN->EST 007 00 015 007 H.323 NO --- NO 00
CS IDLE
7 063-07 023 MCDN->EST 008 00 015 008 H.323 NO --- NO 00
CS IDLE
8 063-08 023 MCDN->EST 009 00 015 009 H.323 NO --- NO 00
CS IDLE

```

vtrkShowByTN

vtrkShowByTN <0x<TN>>

Display the status of the specified trunk, where:

<TN> is the packed TN of the trunk in hexadecimal format.

```

-> vtrkShowByTN 0x00600C
INDEX TN      DCH      CHID CUST ROUTE MEMBER VoIP ESN5
STATUS
0 063-00 023 MCDN->EST 001 00 015 001 H.323 NO CS IDLE

```

DCHmenu

DCHmenu

Display a menu of DCH diagnostic tools.

```

-> DCHmenu
Please select one of the DCHmenu options:
0 - Print menu (default)
1 - Print current DCH state
2 - Print current DCH configuration
3 - Print application error log
4 - Print link error log
5 - Print protocol error log

```

- 6 - Print message log
 - 7 - Enable printing all messages processed by UIPC
 - 8 - Enable error printing
 - 9 - Enable info printing
 - 10 - Enter manual message mode
 - 11 - Print b channel control blocks
 - 99 - Exit menu
- Please enter your DCHmenu choice (0 to print the menu):

DCHdisable

DCHdisable

Disable the D-channel.

DCHenable

DCHenable

Enable the D-channel.

Vtrk message monitor

Display the states and events of the virtual trunk during a call.

syslogLevelSet

syslogLevelSet <tid> or <taskname>, <level>

For the task identified by <tid> or <taskname>, set the log level to the value specified by <level>. The level can be set to a number in the range of 0 to 7. The default value is 6. For more details, see “syslogLevelSet <task>, <level>” on [page 127](#).

-> syslogLevelSet tVTK, 7

uipc_debug_print_on = <uipcFlagState>**uipc_debug_print_on = <UIPCflagState>**

Activate or deactivate the printing of UIPC messages where:

<UIPCflagState> activates or deactivates the printing. Values are:

0—off

1—on

Checking the ELAN link

Use the **pbxLinkShow** to check the status of the ELAN link.

pbxLinkShow

Display the status of the ELAN link between the Signaling Server and the Call Server. The IP address is also displayed.

-> **pbxLinkShow**

Active CS type = Succession CSE 1K

Active CS S/W Release = 201H

Supported Features: GetCSVsn TCP ShiftKey I2050 I2002 CorpDir UserKey-Label VirtualOffice

UseCSPwd

CS Main: ip = 47.11.217.111, ConnectID = 0xc629d64, BroadcastID = 0xc629e60,

Link is up

CS Redundant: ip = 0.0.0.0, ConnectID = 0x0, BroadcastID = 0x0, Link is initializing

CS Signaling Port = 15000

CS Broadcast Port = 15001

Broadcast PortID = 0xc629fe8

RUDP portID = 0xc629f5c

Tcp Link state = up

Tcp Signaling Port: 15000

Tcp socket fd: 26

Tcp msgs sent: 29587

Tcp msgs recd: 1455210

Obtaining information about the Signaling Server

Use the following command to obtain information about the Signaling Server.

itgCardShow

Display information about the Signaling Server, including the uptime. The uptime indicates how much time has elapsed since the last Signaling Server reboot.

```
-> itgCardShow
Index : 1
Type : EXUT
Role : Leader
Node : 8517
Leader IP : 47.11.217.106
Card IP : 47.11.217.107
Card TN : Slot 0
Card State : UNEQ
Uptime : 3 days, 3 hours, 58 mins, 45 secs (273525 secs)
Codecs : G711Ulaw(default), G711Alaw, G729A, G729AB, G723_5, G711CC,
T38FAX
InPci stat : (not implemented yet: Unknown)
```

Network Protocol Module

Network Protocol Module (NPM) is the software module that provides the interface between the H.323 network and the VTRK call control module (IGCC).

NPM interprets incoming H.323 messages from an IP network and converts them into functional call control messages. NPM also converts outgoing functional call control messages into the appropriate H.323 messages.

Call Server MCDN messages are tunneled by NPM in the non-standard data field of the H.323 messages. This tunneling provides an end-to-end MCDN signaling path over the IP network.

NPM also communicates with the H.323 Gatekeeper for registration and admission control.

The following sections describe NPM reporting and debugging commands.

npmShow

Display basic NPM and call status.

-> **npmShow**

```

Npm status:           Active
Active GateKeeper:   47.11.249.140 (primary)
GateKeeper registration status: registered, TTL: 295 secs, re-register: 20 secs
Channels Busy / Idle / Total:  4 / 196 / 200
Stack version:       RadVision 3.0.9.5
Channel tracing:     -1
  
```

Chan	Direction	CallState	RxState	TxState	Codec	AirTime	FS	MS
Fax	DestNum	RemoteIP						
1	Terminate	Offering	-na-	-na-	-none-	8 yes s	no	5100
47.11.249.70								
2	Terminate	Connected	Connected	Connected	G_711_u_low_20MS_NOVAD	5 yes m	no	5101
47.11.249.70								
3	Originate	RingBack	-na-	-na-	-none-	15 no s	no	5701
47.11.215.202								
4	Originate	Connected	Connected	Connected	G_711_u_low_20MS_NOVAD	4 yes s	no	1003
47.11.249.65								

NpmTraceID = <channelNum>

Activate or deactivate NPM tracing on the specified channel number where:

<channelNum> is the channel number. Use the same ISDN channel number that is used on the Call Server. Values range from 1 to 200 for a specific channel number. The value -1 will activate the trace for all channels and the value 0 will deactivate the trace.

-> **NpmTraceID = 1**

NPM debug message logging

NPM uses the syslog facility to write debug log messages to the report system. The NPM debug messages from the **cmHook** send and receive functions are very useful for tracing the flow of H.323 messages. These

routines are low level hook functions called by the H.323 stack as messages are sent or received on the IP network.

To enable NPM debug logging, enter:

```
-> syslogLevelSet tNPM,7
```

To disable NPM debug logging, enter:

```
-> syslogLevelSet tNPM,7
```

The following debug log is from a simple outgoing call.

```
->
-> syslogLevelSet tNPM,7
->
20/06/02 12:54:52 LOG0007 NPM: Service msg q... (1)
20/06/02 12:54:52 LOG0007 NPM: ISDN Call Setup msg rcvd: msgP-
tr=0xc0d9b0c
20/06/02 12:54:52 LOG0007 NPM: npmOutCallSetup: SESSION 0xbf3ca74,
chid 4; uu_msg_length 82
20/06/02 12:54:52 LOG0007 NPM: npmOutCallSetup: Called 1003; Calling
5100
20/06/02 12:54:52 LOG0007 NPM: CHID 4: npmOutCallSetup: Calling num:
5100, Called num: 1003, ESN5 prefix
20/06/02 12:54:52 LOG0007 NPM: npmTunneledUIPEInsert: chid 4 ISDN set-
up: msg 0xc0d9b0c length 82
20/06/02 12:54:52 LOG0007 NPM: cmHookSendTo: 47.11.249.140(1719): RAS:
admissionRequest: hConn 0xc063a2c
20/06/02 12:54:52 LOG0007 NPM: rasSession(0xcbff854), Sent ARQ to Gate-
Keeper
20/06/02 12:54:52 LOG0007 NPM: Handle rad events (timer 5000)...
20/06/02 12:54:52 LOG0007 NPM: Handle rad events (timer 5000)...
20/06/02 12:54:52 LOG0007 NPM: cmHookRecvFrom: 47.11.249.140(1719):
RAS requestInProgress hCon 0xc063a2c
20/06/02 12:54:52 LOG0007 NPM: Handle rad events (timer 4983)...
20/06/02 12:54:52 LOG0007 NPM: Handle rad events (timer 4983)...
20/06/02 12:54:52 LOG0007 NPM: cmHookRecvFrom: 47.11.249.140(1719):
RAS admissionConfirm hCon 0xc063a2c
20/06/02 12:54:52 LOG0007 NPM: cmEvRASConfirm: GK Admission con-
firmed
20/06/02 12:54:52 LOG0007 NPM: cmEvRASConfirm(ACF): Dest CS address
```


47.11.249.65:1720

```

-----
20/06/02 12:54:52 LOG0007 NPM: cmHookSend: 47.11.249.65(1720): Q931: set-
up: hConn 0xc08a870
20/06/02 12:54:52 LOG0007 NPM: Handle rad events (timer 14984)...
20/06/02 12:54:52 LOG0007 NPM: cmHookRecv: Q931: callProceeding: hConn
0xc08a870
20/06/02 12:54:52 LOG0007 NPM: cmHookRecv: FS IE 0 Status in CALL PRO-
CEEDING
20/06/02 12:54:52 LOG0007 NPM: cmEvCallStateChanged: chid 4: Proceeding
20/06/02 12:54:52 LOG0007 NPM: Handle rad events (timer 5000000)...
20/06/02 12:54:52 LOG0007 NPM: cmHookRecv: Q931: alerting: hConn
0xc08a870
20/06/02 12:54:52 LOG0007 NPM: cmHookRecv: FS IE 0: PI IE 0
20/06/02 12:54:52 LOG0007 NPM: cmEvCallStateChanged: chid 4: Ringback:
local
20/06/02 12:54:52 LOG0007 NPM: npmEndpointIdValid: productId is
Nortel_CSE_1000
20/06/02 12:54:52 LOG0007 NPM: npmEndpointIdValid: versionId is 1.0.x
20/06/02 12:54:52 LOG0007 NPM: npmInCodecSelectionVersion: 9
20/06/02 12:54:52 LOG0007 NPM: npmSendRingbackReq: sending PBXUser-
Input - ringback on
20/06/02 12:54:52 LOG0007 NPM: npmTunneledUIPEExtract: ISDN alerting
msg 0xc18c498, length 18
20/06/02 12:54:52 LOG0007 NPM: npmInCallAccept: sending IgccNpmCallAc-
cept message
->
-> syslogLevelSet tNPM,6

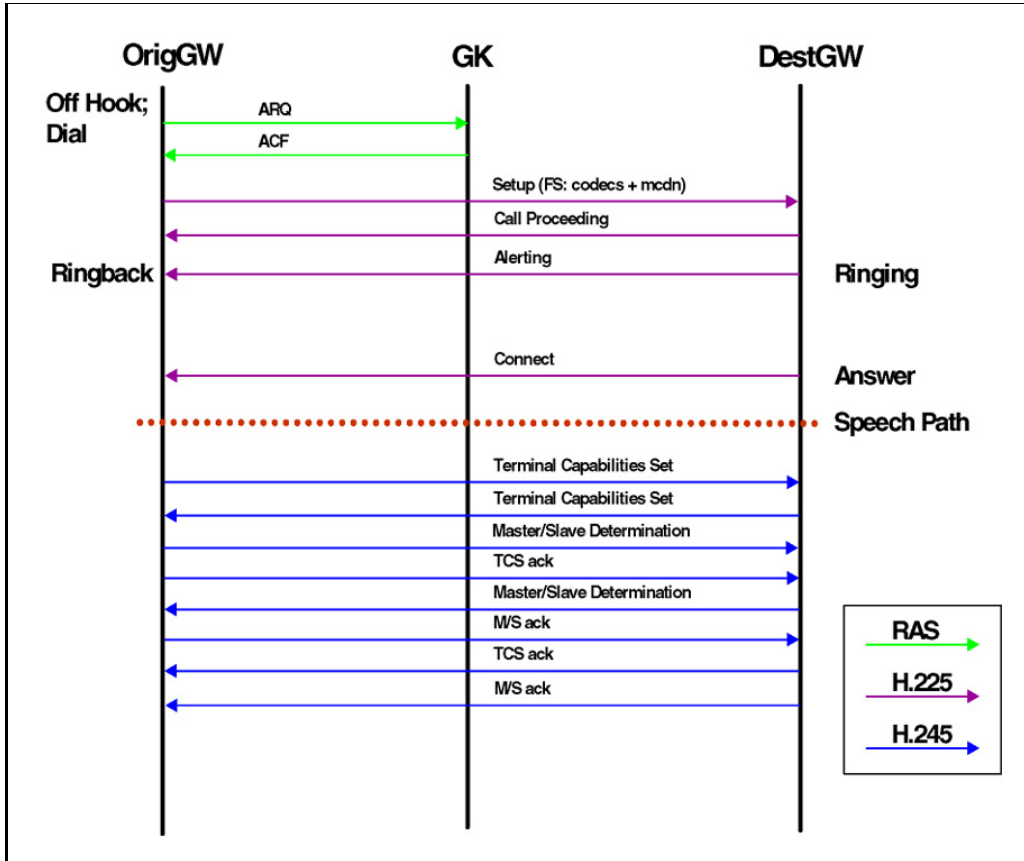
```

NPM H.323 message flow—originating a simple call

Figure 8: “H.323 message flow—originating a simple call” on [page 186](#) illustrates the H.323 message flow between the OrigGW and DestGW endpoints during the origination and establishment of a simple call. The diagram also illustrates the endpoint interaction with the Gatekeeper.

Note: You can correlate the messages in the debug log of the preceding example to the flow diagram.

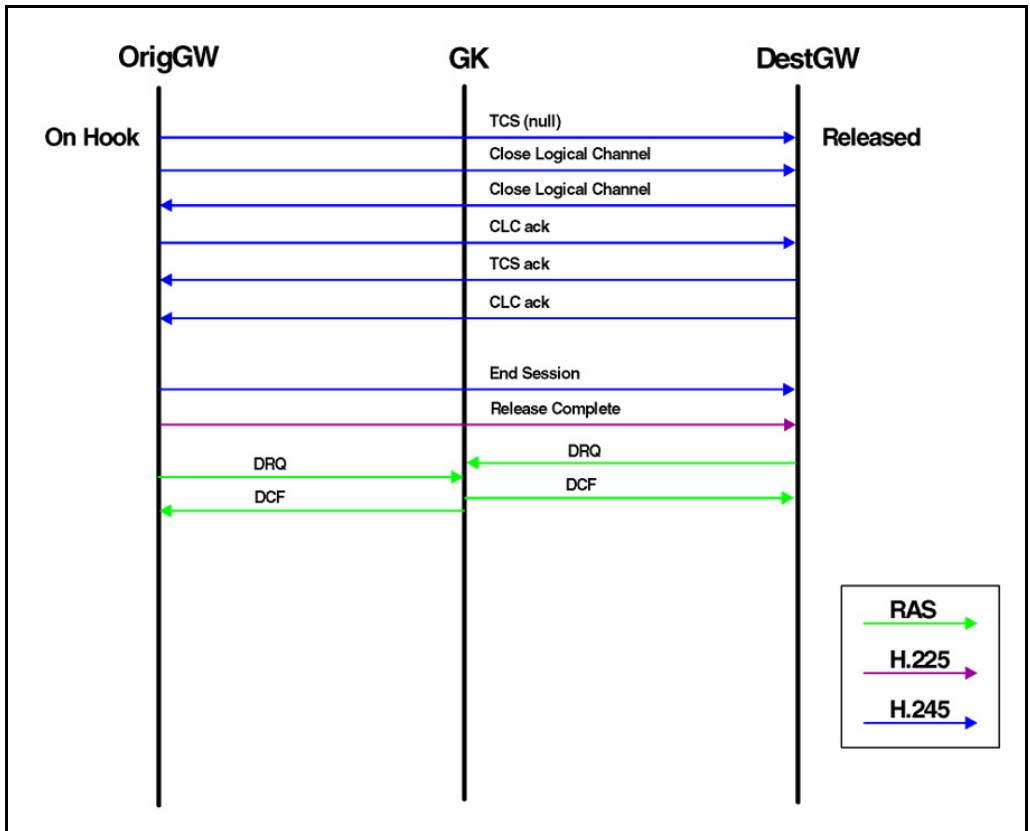
Figure 8
H.323 message flow—originating a simple call



NPM H.323 message flow—releasing a simple call

Figure 9: “H.323 message flow—releasing a simple call” on [page 187](#) illustrates the H.323 message flow that occurs when a simple call is released.

Figure 9
H.323 message flow—releasing a simple call



Signaling Server installation

This chapter provides details on Signaling Server installation procedures and files from a troubleshooting perspective.

The following NTPs contain further details about Signaling Server installation, configuration, and upgrade procedures:

- *Communication Server 1000M and Meridian 1: Large System Installation and Configuration (553-3021-210),*
- *Communication Server 1000M and Meridian 1: Large System Upgrade Procedures (553-3021-258),*
- *Communication Server 1000S: Installation and Configuration (553-3031-210),*
- *Communication Server 1000S: Upgrade Procedures (553-3031-258),*
- *Communication Server 1000E: Installation and Configuration (553-3041-210),*
- *Communication Server 1000E: Upgrade Procedures (553-3041-258),* and
- *Signaling Server: Installation and Configuration (553-3001-212).*

Directory structure on the software CDROM

The following is the directory structure on the Software CDROM of Signaling Server:

- The root directory of CDROM contains the boot files.
- The subdirectory **LOAD/** contains the Install Tool object file.

- The subdirectory **SSExxxx.P3/** contains the SS software.
- File **IPLxxxx.P2** contains the IP Line loadware for the ITG Pentium 2 card (ITG-L or ITG-T).
- File **IPLxxxx.SA** contains the IP Line loadware for the ITG StrongArm card (Succession Media Card).
- File **0602Bxx.BIN** contains the internet telephone firmware for i2004 telephones.
- File **0603Bxx.BIN** contains the internet telephone firmware for i2002 telephones.
- The subdirectory **SYMTABS/** contains symbol tables used for debugging.
- If it exists, subdirectory **PATCHES/** contains the SS patches.
- The subdirectory **MKBOOT/** contains the mkboot utility and files.

The CDROM image is as per the ISO 9660 specification, is bootable according to the El Torito specification, and uses "floppy emulation". Your CDROM burner software creates the CD using an "ISO" image file. See section "Signaling Server boot-up" on [page 17](#) for more information.

The mkboot utility creates the boot floppy for regular boot to hard disk, or bootable CDROM.

The install index file (**INDEX.DAT**) lists the software files on the software CDROM. This file is automatically created during loadbuild.

The install control file (**INSTALL.DAT**) lists the software version, the directories, files, and checksums (a checksum is not calculated for **INSTALL.DAT**). The install control file is automatically generated at loadbuild.

To manually calculate the checksum, use the utility **/proj/cseSys/target/config/tools/checksum**.

Disk LayOut (DLO) and DISK.SYS

Install Tool only contains basic data, therefore, a preliminary version of **DISK.SYS** is installed during the Install Tool initialization to provide the necessary information for Install Tool operation.

A temporary version of **DISK.SYS** is installed after a software load is selected. The temporary version is removed at the end of installation process. The final version of **DISK.SYS** is installed on **/p** as part of installation.

The DLO search order is: **/u -> /p -> /cd0 -> /f0**. Use the command **dloShow** to view options.

Install Tool initialization

During Install Tool initialization, the system verifies the file systems, partitions the hard disk if necessary, and reboots itself.

Note: A hard disk controller failure can cause partitioning to fail.

The following example is from a new system installation.

Verifying filesystems ...

Succession CSE 1000 Signaling Server CDROM Install Tool (sse-2.00.72)

```
=====
=====
```

The filesystems verification failed! (This is normal for a new system.)

The hard disk must be (re)partitioned and (re)initialized. This will erase all data on the hard disk. The system will then reboot and the Install Tool will restart.

Please enter:

<CR> -> <a> - Partition and initialize the hard disk, then reboot.

Enter Choice><CR>

Partitioning hard disk ...
Configuring disk hda
ataPartCreate(0, 0, 1, 6, 2048)
ataPartTableExtend(0, 0, f, 0)
ataPartCreate(0, 0, 5, 6, 2048)
ataPartTableExtend(0, 0, 5, 0)
ataPartCreate(0, 0, 9, 6, 2048)
ataPartCreate(0, 0, 10, c0, 1024)
Hard disk partitioning succeeded.

Creating filesystems ...
Filesystems creation succeeded.

Rebooting system ...

After a successful file system verification, the preliminary **DISK.SYS** file is installed. If there is no existing CS software (check **/p/data/version.dat**), the system prompts you to enter the current date and time.

Verifying filesystems ...
Filesystems verification succeeded.

Copying "/cd0/disk.sys" to "/u/disk.sys".

Succession CSE 1000 Signaling Server CDROM Install Tool (sse-2.00.72)

=====

You should ensure that the system date and time are correct prior to installation, since all files copied or created during install will be time-stamped.

You can press <CR> to accept the current values.

Current date is: WEDNESDAY 16-10-2002
Enter new date (dd mm yyyy):<CR>
Date not changed.

Current time is: 15:05:02
Enter new time (hh mm ss):<CR>
Time not changed.

**Current date and time is:
WEDNESDAY 16-10-2002, 15:05:02**

The system performs a disk test, if necessary, or at your request. In this example, the hard disk test is mandatory.

Succession CSE 1000 Signaling Server CDROM Install Tool (sse-2.00.72)

=====

The Install Tool cannot determine when the hard disk was last tested.

The hard disk must be tested before installation can continue. This test will take approximately 14 minutes.

**Please enter:
<CR> -> <a> - Test the hard disk.**

Enter Choice><CR>

**Testing hard disk ...
Testing partition /u (4194241 blocks) ...
100% complete**

**Testing partition /p (4194241 blocks) ...
100% complete**

Hard disk testing succeeded.

Note: The hard disk test must be successful before installation can proceed.

Hard disk partitioning

The hard disk partitioning is similar to that used in the CPP file system. It is not 100% compatible with the DOS file system (fdisk). Large Block Addressing (LBA) is not supported.

The hard disk partition structure contains:

- a primary partition (**/p**)—a 2GB “protected” partition,
- an extended partition,
- a logical partition (**/u**),—a 2GB "unprotected" partition.

Use the **ataPartTableShow** to display the hard disk partition information.

Disk testing

If the disk test log file does not exist, the hard disk test is performed. This will occur under the following conditions:

- the system is a new system
- the system is reformatted
- the previous disk test failed or was aborted

If the disk test log file is < 24 hours old, the default action is to skip the hard disk test.

If the disk test log file is >= 24 hours old, the default action is to test the hard disk.

You can bypass the disk test at any time before Install Tool displays the disk test menu, by following this procedure:

- 1 Access vxshell (via the front maintenance port).
- 2 Manually create a disk test log file (copy **>/u/disktest.log**).
- 3 Type anything (or nothing) into this file.
- 4 End the file (CTRL-D).
- 5 Exit vxshell.
- 6 Continue with the execution of Install Tool.

Tested file systems

The devices that are auto-mounted (those without the "noauto" option) are tested.

Use the **mtabShow** command to display the file system information:

```
-> mtabShow
```

Device	BlkDev	Mount Path	DevHdr	Mount Options
/dev/hda1	0x0bcc4594	/p	0x0cb36764	ro,autosync
/dev/hda5	0x0bb58684	/u	0x0cb36560	rw,autosync
/dev/hdc1	0x0bb37324	/cd0	0x0bb372d0	user,noauto,ro
/dev/fd0	0x0bb357cc	/f0	0x0bb667b8	user,noauto,rw,changeswarn

Disk test functions

The disk test reads all blocks in the disk partitions and reports any read failures. If a failure occurs, installation halts.

Note: A disk controller failure also causes the disk test to fail.

The following example shows the system output for a disk test failure.

```
Testing hard disk ...
Testing partition /u (4194241 blocks) ...
100% complete
```

```
Testing partition /p (4194241 blocks) ...
Problem with blocks at 3063808!
```

```
INST0173 Fatal, fail to test the hard disk.
```

```
Succession CSE 1000 Signaling Server CDROM Install Tool (sse-2.00.70)
```

```
=====
```

```
The hard disk test failed!
```

```
You may have to replace your server unit. Please contact your
technical support group.
```

```
Software installation cannot continue!
```

Signaling Server software installation

The system performs the following steps during Signaling Server software installation:

- The system compares the software version of the new software being installed with the existing software version (**/p/data/version.dat**). If the **version.dat** file does not exist, the system considers the installation to be a new software installation.
- The system prompts you to begin the software installation.
- The system backs up the file **/p/nvram.sys**, if it exists, to **/u/nvram.bak**.
- The system erases the protected partition (**/p**).
- The system creates the necessary directories, if they don't already exist.
- The system copies each file, block by block, and calculates a checksum.
- The system restores the **/p/nvram.sys** file.
- The system installs the boot track and the boot loader (**bootrom.sys**).
- The system erases the old patches if the software is a new version.
- The system prints an Installation Status Summary.

INSTALLATION STATUS SUMMARY

	Option		Choice		Status		Comment	
	software		yes		ok		new install 2.00.72	
	firmware		yes		ok		copy i2002 version 1.39	
	firmware		yes		ok		copy i2004 version 1.39	
	loadware		yes		ok		copy IP Line 3.00.72 for P2	
	loadware		yes		ok		copy IP Line 3.00.72 for SA	

```

+-----+-----+-----+-----+
| configuration | yes | ok | set as Leader |
+-----+-----+-----+-----+
    
```

VGMC loadware copy

The system copies selected loadware files to the destination directory, **/u/fw**, on the SS hard disk (per DLO) and prints the Installation Status Summary.

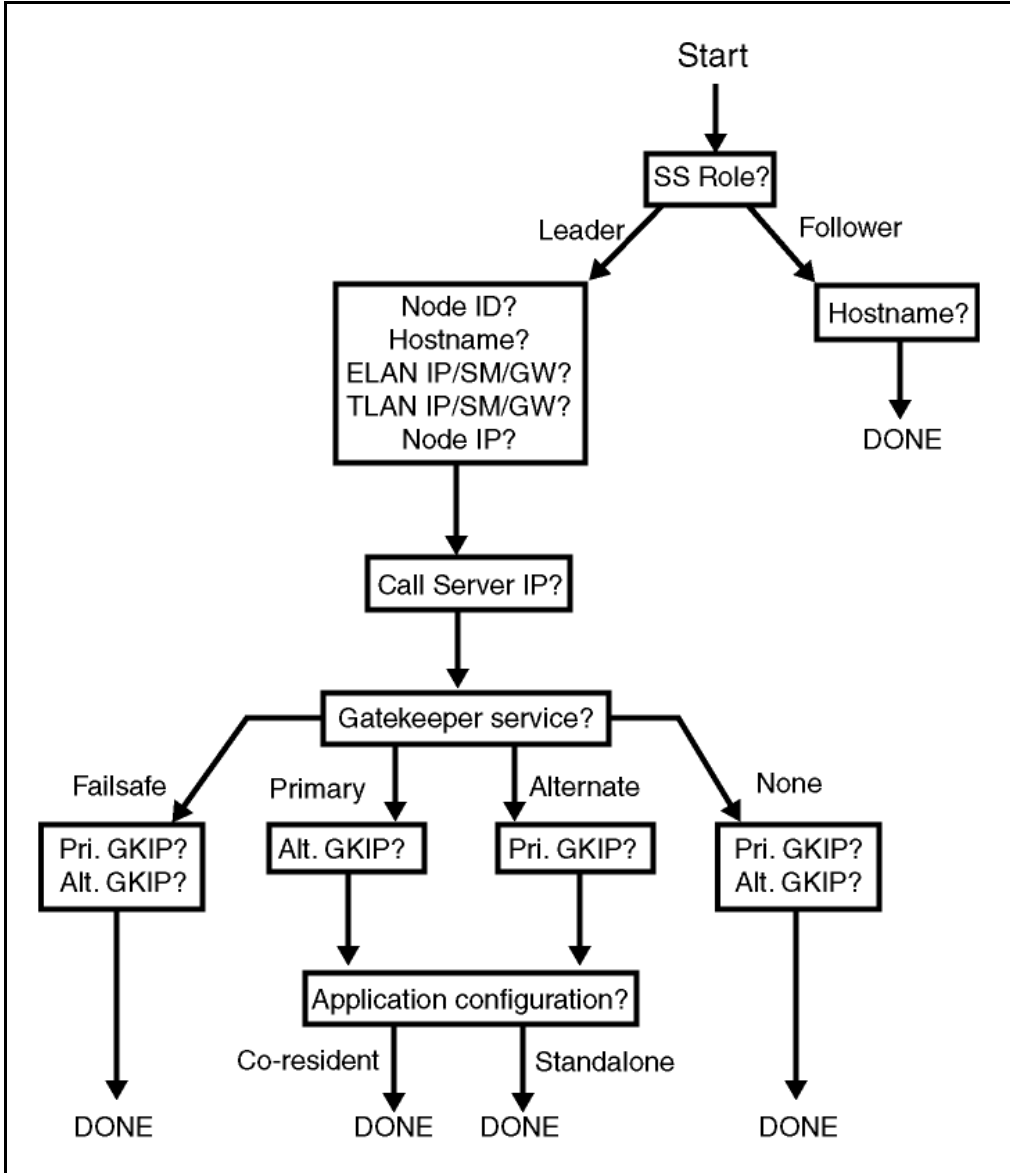
Internet Telephone firmware copy

The system copies selected firmware files to the destination directory, **/u/fw**, on the SS hard disk (per DLO) and prints the Installation Status Summary.

Basic configuration

Figure 10: “Basic configuration of the Signaling Server” on [page 198](#) provides a pictorial representation of the different options available during the configuration process.

Figure 10
Basic configuration of the Signaling Server



Leader configuration

You must enter the following parameters to configure a leader Signaling Server:

- Node id
- My hostname
- My ELAN parameters
- My TLAN parameters
- Node IP
- Call Server IP
- Gatekeeper role (Primary/Alternate/Failsafe/None)
- Configuration, if applicable (standalone, co-resident).
- Gatekeeper IPs (may or may not be optional, depending on the GK role)

After the you enter the parameters, the Signaling Server performs the following steps to complete the leader configuration:

- writes a new BOOTP.TAB (with my network parameters only)
- writes a new NVRAM.SYS (boot parameters file), with flags=0x2000
- writes new CONFIG.INI (with my application parameters only)

Follower configuration

You must enter the following parameter to configure a follower SS:

- my hostname

After the you enter the parameter, the Signaling Server performs the following steps to complete the leader configuration:

- display the ELAN MAC address.
- writes a new **nvr_{am}.sys** (boot parameters file) with **flags=0x0**.
- writes the default **config.ini** (no application configuration).

The follower SS obtains the network parameters via BOOTP, and the new node files via FTP, from the leader SS.

Changing the Web Server security flag

By default, Element Manager can be accessed from management workstations (web browsers) on any subnet, however, the Web Server security flag can be set to restrict access to EM. Only hosts on the ELAN subnet can access EM if the Web Server security flag is set.

Setting the Web Server security flag:

- 1 Select option **e** on the SS Install Tool menu.
The system scans the **config.ini** file for the Web Server security flag and displays the current value. (The default value is FALSE).
The system issues a prompt asking for a new value.
- 2 Enter a new value for the Web Server security flag.
The system updates the parameter value in **config.ini**.
- 3 Re-import the Element Manager **config.ini** file to update the master copy on the Call Server.

Changing the Signaling Server configuration

The Signaling Server (excluding the Gatekeeper) uses three configuration files:

- **/p/nvram.sys**, containing the boot parameters, including the ELAN parameters,
- **/u/config/bootp.tab**, containing the TLAN parameters, and
- **/u/config/config.ini**, containing the VoIP configuration, including the application configuration, and the Call Server and Gatekeeper IP addresses.

Changing the network configuration

There are two methods to change the network configuration (more specifically, the ELAN or TLAN IP addresses).

Method one

This method involves a single step:

- 1 Configure the new boot parameters using the Install Tool with the “configuration only” option.
The system overwrites the existing configuration files.

Method two

This method consists of two steps:

- 1 Enter the changed network configuration parameters using the edit node configuration screen in Element Manager.
- 2 Perform the transfer operation to download the new **bootp.tab** and **config.ini** files to the Call Server.

Changing the application configuration

To change the application configuration, including the Call Server and Gatekeeper IP addresses:

- 1 Enter the changed network configuration parameters using the **edit node** configuration screen in Element Manager.
- 2 Download the new **bootp.tab** and **config.ini** files to the node.

Troubleshooting Problem Conditions

The following tables identify causes and solutions of common problems.

**Table 5
Troubleshooting Signaling Server software installation problems**

Symptoms	Possible Causes	Solutions
- cannot see anything on maintenance terminal during boot	- incorrect serial cable - incorrect port speed	- use a DTE-DTE null-modem serial cable between maintenance port and terminal - default port speed is 19200
- cannot see AMIBIOS boot screen	- hardware flow control is not available	- use a full serial cable between maintenance port and terminal - enable hardware flow control on terminal

Table 5
Troubleshooting Signaling Server software installation problems

Symptoms	Possible Causes	Solutions
- cannot press F2 to enter BIOS setup	- some terminals don't pass the function keys	- connect a PC keyboard directly to the Signaling Server - try a different terminal / emulation
- system does not boot at all with software CDROM inserted	- software CDROM is not bootable or corrupt	- try another software CD-ROM - verify that directions were followed to create software CDROM (see README.TXT that came with ISO file) - create and use boot floppy (see mkboot)
- system does not boot from boot floppy	- boot floppy is not bootable or corrupt - BIOS boot order is unsuitable	- try another boot floppy - verify that directions were followed to create boot floppy (see mkboot) - change BIOS boot order so that floppy is searched first
- with software CDROM inserted, system still boots to hard disk	- "hard disk" was selected at boot menu - boot menu timed out (default: hard disk) - BIOS boot order is incorrect	- at boot menu, enter "c" to boot from CDROM - reset BIOS defaults
- INST0173 Fatal, fail to test the hard disk.	- the hard disk has bad blocks - the hard disk controller is faulty	- replace Signaling Server unit

Table 5
Troubleshooting Signaling Server software installation problems

Symptoms	Possible Causes	Solutions
<ul style="list-style-type: none"> - No such file or directory. - INST0037 Unable to open file "X" for reading. - INST0014 Error parsing the install control file. - INST0011 Unable to process the install control file. - INST0019 Unable to make directory "X". - INST0012 Unable to open file "X". - INST0027 Error copying file "X". - INST0029 Checksum of "X" does not match. 	<ul style="list-style-type: none"> - the software CDROM is corrupt - the hard disk is corrupt 	<ul style="list-style-type: none"> - try installation, or reading the file, again - try another Software CDROM - repartition and reformat the hard disk, then reinstall
<ul style="list-style-type: none"> - cannot install Internet Telephone firmware - cannot install VGMC loadware - cannot perform basic Signaling Server configuration - cannot change the web server security flag 	<ul style="list-style-type: none"> - software or configuration directories don't exist - the hard disk is corrupt 	<ul style="list-style-type: none"> - first install the Signaling Server software - repartition and reformat the hard disk, then reinstall

Table 6
Troubleshooting Signaling Server configuration problems

Symptoms	Possible Causes	Solutions
- Signaling Server does not boot	<ul style="list-style-type: none"> - invalid CDROM or boot floppy are in drives - BIOS boot order is incorrect - software has not been installed - configuration is incorrect 	<ul style="list-style-type: none"> - remove CDROM or boot floppy from drives - reset BIOS defaults - (re)install the software - verify Signaling Server's IP telephony configuration
- cannot ping ELAN	<ul style="list-style-type: none"> - Signaling Server has not booted successfully - data network configuration / routing problem 	<ul style="list-style-type: none"> - verify that Signaling Server has booted successfully - verify Signaling Server data network configuration and connectivity
- cannot ping TLAN	<ul style="list-style-type: none"> - Signaling Server has not booted successfully - data network configuration / routing problem - application configuration / routing problem 	<ul style="list-style-type: none"> - verify that Signaling Server has booted successfully - verify Signaling Server data network configuration and connectivity - verify Signaling Server's IP telephony configuration

Table 6
Troubleshooting Signaling Server configuration problems

Symptoms	Possible Causes	Solutions
<ul style="list-style-type: none"> - telephones don't register 	<ul style="list-style-type: none"> - Signaling Server is not available - incorrect IP telephony parameters (node ID, node IP) - incorrect system configuration 	<ul style="list-style-type: none"> - verify that Signaling Server has booted successfully - verify data network connectivity between telephones and SS - verify that Signaling Server is running TPS - verify Signaling Server's IP telephony configuration - verify telephones' IP telephony configuration - verify system TN configuration (LD 11)
<ul style="list-style-type: none"> - cannot make outgoing trunk calls 	<ul style="list-style-type: none"> - virtual trunk and/or GK configuration is corrupt 	<ul style="list-style-type: none"> - verify configuration in Element Manager - (re)transfer node files from Element Manager
<ul style="list-style-type: none"> - cannot login to CLI 	<ul style="list-style-type: none"> - incorrect login or password 	<ul style="list-style-type: none"> - try using the system's PWD2 - try using the default administrator login and password - reset the administrator login and password using the Install Tool

Table 6
Troubleshooting Signaling Server configuration problems

Symptoms	Possible Causes	Solutions
<ul style="list-style-type: none"> - cannot access Element Manager web pages 	<ul style="list-style-type: none"> - Signaling Server is not available - web server has not loaded - there is no route between Signaling Server and browser PC - web server security flag is enabled and browser PC is not on ELAN 	<ul style="list-style-type: none"> - verify that Signaling Server has booted successfully - verify data network configuration and connectivity - verify browser PC and Signaling Server configuration - disable web server security flag, or move browser PC to ELAN subnet
<ul style="list-style-type: none"> - cannot login to Element Manager 	<ul style="list-style-type: none"> - incorrect login or password or Call Server IP - Login Names option is disabled - Call Server is unavailable - login is already/ still in use - logins are blocked, due to too many incorrect logins - browser is caching old pages - incorrect browser used 	<ul style="list-style-type: none"> - verify login, password, Call Server IP - enable login names (LD 17) - verify that Call Server is available - logout existing user, or wait for user to time out automatically - wait for logins to unblock automatically, or reboot the Call Server - disable browser caching (set to "reload pages every time") - use only Microsoft IE 5.5 or higher

Table 7
Troubleshooting Call Server and Media Gateway software installation problems

Symptoms	Possible Causes	Solutions
<ul style="list-style-type: none"> - failed to load DRAMOS from software delivery card - cannot read PC card 	<ul style="list-style-type: none"> - software delivery card was formatted using Windows 2000 - software delivery card was not programmed correctly 	<ul style="list-style-type: none"> - try another PC card - reformat PC card from PDT using pcmcia format command
<ul style="list-style-type: none"> - cannot return or quit in software installation program menus - software installation program cursor is not at end of prompt 	<ul style="list-style-type: none"> - INSTALL.LST is corrupt, likely due to Unix to DOS text conversion during FTP of files to PC card 	<ul style="list-style-type: none"> - FTP the files again, using BIN mode
<ul style="list-style-type: none"> - found unerasable track 	<ul style="list-style-type: none"> - corrupt file system, usually due to a reboot at a bad time, such as during boot or installation 	<ul style="list-style-type: none"> - use dosFsCheck c:,1,1 to repair file system - use FBUG to erase and test flash drive on daughterboard - try a different software daughterboard
<ul style="list-style-type: none"> - software installation is very slow 		<ul style="list-style-type: none"> - use FBUG to erase and test flash daughterboard, then reinstall software - try a different software daughterboard

Table 8
Troubleshooting SIPE or centralized software upgrade problems

Symptoms	Possible Causes	Solutions
<ul style="list-style-type: none"> - invalid opcode -continuous warm starts early in boot sequence 	<ul style="list-style-type: none"> - incorrect symbol file is loaded from software delivery card 	<ul style="list-style-type: none"> - remove software delivery card from Media Gateway during CSU
<ul style="list-style-type: none"> - security device not installed. System ID read failed. - SRPT063 wrong/ missing dongle detected in cabinet, upgrade aborted. - Media Gateways appear to be in survival mode, but they are not configured for survivability. - SRPT 1013 STARTUP: cabinet x registration denied: system IDs do not match. 	<ul style="list-style-type: none"> - dongle read failed on CS or MG 	<ul style="list-style-type: none"> - reboot SSC to try again
<ul style="list-style-type: none"> - BUG111, GW number illegal, aborted. 		<ul style="list-style-type: none"> -reboot SSC to try again
<ul style="list-style-type: none"> - multiple time-outs and retries during installation - slow upgrade 	<ul style="list-style-type: none"> - data network or link problem - flash program store is slow to program - flash drive has corruption 	<ul style="list-style-type: none"> - correct any data network link problem (switch is auto-negotiate? link is 100BaseT full duplex?) - use FBUG to erase and test daughterboard - try a different software daughterboard

Table 8
Troubleshooting SIPE or centralized software upgrade problems

Symptoms	Possible Causes	Solutions
- CRC errors	- program store could not be programmed correctly - new software MIB is corrupt	- attempt upgrade again - use FBUG to erase and test flash program store on daughterboard - try a software daughterboard - try a different software delivery card
- TFTP server: could not send client file... Transfer Timed Out	- TFTP server is verbose!	- not a problem, ignore
- SRPT063 RPC call failed to invoke upgrade	- RPC mechanism reports failure, but message probably succeeded	- not a problem, ignore
- SRPT1027 STARTUP: IP port is not 100BaseT Full Duplex	- port is not connected - port is not in 100BaseT full duplex mode	- if there is a cabinet on this port, and it has registered, then there is a link problem that must be fixed as it may cause call processing problems!

Element Manager

This chapter describes the different components of Element Manager on CS1000 Release 4.0 from a troubleshooting perspective. These components include: the Web Browser/Client, the Web Server, the Remote Procedure Call (RPC), the Transaction Server, and the Call Server.

The Web Server runs on the Signaling Server platform. The http daemons are created on the SS.

The Web Browser/Client, typically an IE 5.5 +, interfaces with the Web Server through the network.

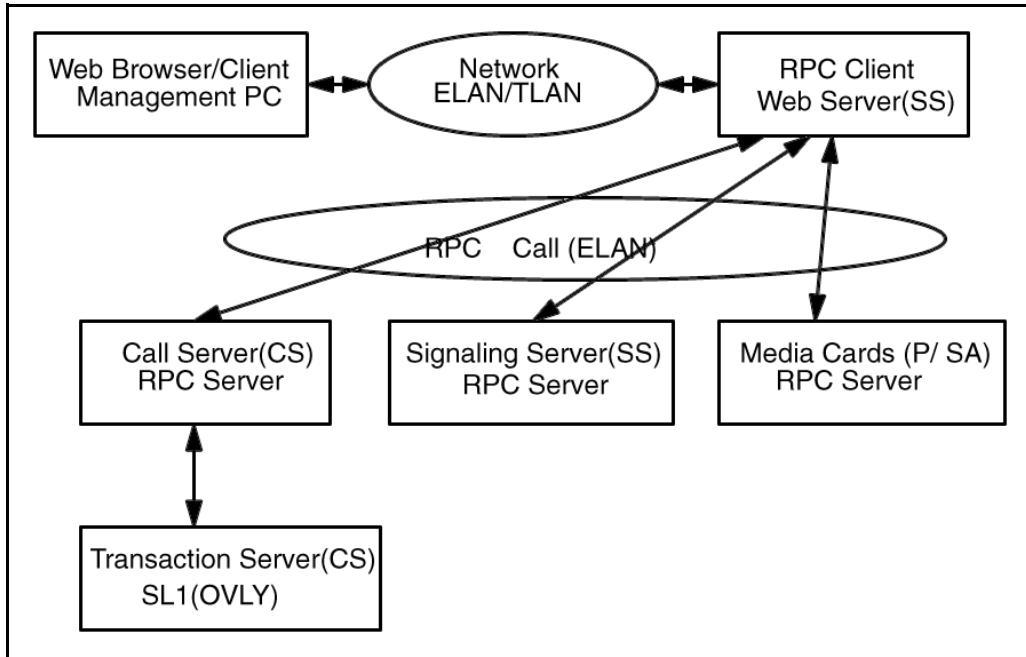
The RPC is a messaging mechanism operating between the Web Server and the Call Server, Signaling Server, or Media Cards (ITG-P or SA).

The Transaction Server is an intermediate layer between the RPC and the Call Server. The Transaction Server validates the incoming and outgoing transactions of the Call Server.

A typical transaction from EM uses all of the subsystems and components mentioned in this section. Figure 11: “Element Manager components” on [page 212](#) illustrates how these components are linked together.

When an overlay transaction or a submit of IP telephony configuration data fails, it is crucial to be able to understand which component caused the failure.

Figure 11
Element Manager components



EM components

The following sections provide useful techniques for debugging problems related to the individual EM components.

Note: Each of the EM components has its own limitations. These limitations have been identified at the beginning of the section describing the component. Always check the limitations of a component before beginning the debugging process.

Web Browser/Client

Web Browser/Client limitations

The limitations of the Web Browser/Client component are:

- Supported browsers—Microsoft IE 5.5+
- Unsupported browser—Netscape
- IE configuration requirement—You must select "**Every visit to the page**" at **Tools->Internet Options->Temporary Internet Files->Settings**; otherwise the latest information is not displayed.
- Browser time out— IE (Windows) time-out is between 1.5 and 2 minutes. If a timeout occurs, "**Page Not Found**" is displayed.
- PC recommendation—P III 500+

Web Browser debug tips

General errors

Pay attention to messages displayed on the browser when an error occurs.

If you cannot get to EM System Information page when attempting to log into EM, check the following:

- If the browser displays **WEB4008** ("No idle TTY available"), ensure that at least 2 PTYs are configured on the Call Server side.
- If the browser displays a message indicating that overlay memory is in use, ensure that no other user is accessing the same overlay from the tty, from a rlogin session, or from another EM login running the same transaction.
- If the browser displays **OVL428**, ("login name and password combination is invalid"), ensure that the login name has been configured in overlay 17 (LAPW).
- If the browser displays **WEB4016**, ("Web Client can't access current overlay"), check the LAPW user password list under the navigation menu, **Administration->Password**. This list identifies which overlays each user can access. If you are an LAPW user, you can not add overlays to the password list. Admin1 or Admin2 users only can add overlays to the password list.
- If the browser displays **WEB3003**, ("destination IP address cannot be reached, initial RPC failed"), check the Call Server IP address by pinging the CS IP address from the SS shell.

- If the browser displays **WEB0030**, ("login page has been inactive for too long"), click OK to continue and re-login by entering your userid and password.
- If the browser displays **WEB0019**, ("access to the Web Server is denied"), ensure that the Web Server http tasks are running on the Signaling Server.

Overlay or configuration submit failures

WEB0007, ("incoming CGI query is invalid"), is sometimes displayed during an overlay transaction or during a submit of IP telephony configuration data. This message indicates that the Web Server does not understand the CGI string sent by the client side.

The most probable cause of the problem is the CGI string being chopped into multiple strings. To confirm the cause, set the **httpDebugFlagUtil** debug flag on the web server side. See "Web Server debug tips" on [page 216](#) for more information about how to use debug flags.

Node configuration failures

If the browser displays **WEB2503**, (" Initial RPC failed"), while you are importing a node configuration, check:

- if the leader is running.
- if the leader is a Media Card, and if so, ensure that it is running IPL 4.x software.

If you receive a failure message about BOOTP or /CONFIG while transferring a node configuration from the IP telephony page, do the following:

- Ping the destination card element to ensure that it is alive.
- Issue the **i** command on the card element and ensure that the RPC management tasks are not suspended. The RPC task name is **tRPCMGMT**.

-> i

NAME	ENTRY	TID	PRI	STATUS	PC	SP	ERRNO	DELAY
------	-------	-----	-----	--------	----	----	-------	-------

```
tRPCMGMT _start_ss_se 183ef24 200 PEND 34b078 183ee0c d0003 0
```

- Click **GEN CMD** on the status page of the IP Telephony node to verify that messaging is functioning properly.
- Determine if there are more than five active FTP sessions on the CS side. The transfer process initiates an FTP session between the CS and the card element to transfer the BOOTP and /CONFIG files.
- Ensure that the ELAN and TLAN IP address are correct.

The loadware and firmware upgrade functions on the IP telephony page also use FTP sessions to upgrade the loadware and distribute firmware.

If a loadware upgrade or a firmware distribution, initiated from the IP telephony page, results in failure, do the following:

- Ping the destination card element to ensure that it is alive.
- Issue the **i** command on the destination card element to ensure that the RPC management task is running
- Issue the **i** command on the SS (Web Server) to determine if there are more than four active FTP sessions.

Web Server

Web Server limitations

The limitations of the Web Server component are as follows:


- The 4 HTTP daemons on the Web Server can handle 4 simultaneous calls. There is no limitation on the number of users that can be logged in to Web Server, but if more than 4 users simultaneously request access to the Web Server, the requests will be placed in a queue and may potentially time-out.

- Specific characters, when entered in the **H323 ID**, **System name**, **System location** or **System Contact** field within the **Edit Node** page, may cause problems because the browser side identifies these characters as invalid input. The characters are: '#', '%', '^', '<', '?' and '|'.

Web Server debug tips

You can set the following debug flags on the Signaling Server to aid in troubleshooting the Web Server.

Note: The http daemons only reside on the Signaling Server.



CAUTION — Service Interruption

Setting the debug flags on the Web Server side may sometimes cause a warm start on the Signaling Server.

When the individual debug flags are set, the output of a transaction is saved in a file on the SS. You can use the information stored in the file to determine if the Web Server has received the expected information, and if any errors have occurred. You set the flags directly in the VxWorks shell of the SS. The debug flags are:

- **httpDebugFlagutil = 4**

`/webxml/xmltmp/query.xml` always contains the last CGI information received from a web browser. When troubleshooting, ensure that there are no other users on the system, otherwise the CGI information in the file may be information received from another user's browser. The following example displays the contents of the file immediately after configuring Zone using EM.

```
-> httpDebugFlagutil=4
httpDebugFlagutil = 0x26755c8: value = 4 = 0x4
-> cd "/webxml/xmltmp"
value = 0 = 0x0
-> ll
  size      date      time      name
-----
   512    OCT-16-2002 16:04:36 .      <DIR>
   512    OCT-16-2002 16:04:36 ..     <DIR>
```



```

87 OCT-16-2002 16:04:44 query.xml
value = 0 = 0x0
-> copy < query.xml
u=henry&t=1dcd64ec00000001&h=context/callsvr/zn-
browse?x=zonelist&o=117&COMMAND=PRT ZDESvalue = 0 = 0x0
-> httpDebugFlagutil=0

```

- **httpDebugFlagxfiles = 6**

`/webxml/xmltmp/btp.xml` and `/webxml/xmltmp/cfg.xml` contain the `bootp.xml` and `config.xml` information generated during a Web Browser node submit. The following example shows the contents of the files immediately after submitting a node.

```

-> httpDebugFlagxfiles=6
httpDebugFlagxfiles = 0x26755cc: value = 6 = 0x6
-> cd "/webxml/xmltmp/"
value = 0 = 0x0
-> ll

```

size	date	time	name	
512	OCT-16-2002	16:13:22	.	<DIR>
512	OCT-16-2002	16:13:22	..	<DIR>
547	OCT-16-2002	16:13:22	btp.xml	
4052	OCT-16-2002	16:13:22	cfg.xml	

```

value = 0 = 0x0
-> copy < btp.xml
<?xml version="1.0"?>
<bootp>
<GENERAL>
<VERSION>IPLINE3.0</VERSION>
</GENERAL>
<COMMON>
<ELAN_SM>255.255.254.0</ELAN_SM>
<ELAN_GW>47.11.215.114</ELAN_GW>
<NODE_IP>47.11.215.134</NODE_IP>
<TLAN_SM>255.255.254.0</TLAN_SM>
<NODE_ID>435</NODE_ID>
</COMMON>
<SS_ELEMENT>
<INDEX>1</INDEX>
<MAC>00:02:b3:86:1b:ba</MAC>
<ELAN_IP>47.11.216.194</ELAN_IP>

```

```
<TLAN_IP>47.11.215.126</TLAN_IP>
<TLAN_GW>47.11.215.1</TLAN_GW>
<TYPE>SS</TYPE>
<ROLE>Leader</ROLE>
<TN></TN>
</SS_ELEMENT>
</bootp>
value = 0 = 0x0
-> copy < cfg.xml
<?xml version="1.0"?>
<config>
  <GENERAL>
    <VERSION>IPLINE3.0</VERSION>
  </GENERAL>
  <KEYCODE>
    <KEYCODE_ID>12345678-12345678-12345678</KEYCODE_ID>
  </KEYCODE>
  <SECURITY>
    <ELANACCESSONLY>0</ELANACCESSONLY>
  </SECURITY>
  <SNMP>
    <RD_COMMUNITY_NAME>public</RD_COMMUNITY_NAME>
    <WR_COMMUNITY_NAME>private</WR_COMMUNITY_NAME>
    <TRAPS_ENABLED>1</TRAPS_ENABLED>
    <TS_IP1>0.0.0.0</TS_IP1>
    <TS_SM1>255.255.254.0</TS_SM1>
  </SNMP>
  <SNMP_ELEMENT>
    <ELAN_IP>47.11.216.194</ELAN_IP>
    <SYS_HOSTNAME>Henry Yang</SYS_HOSTNAME>
    <SYS_LOCATION>Bay Terrace 1</SYS_LOCATION>
    <SYS_CONTACT>Henry x35555555</SYS_CONTACT>
  </SNMP_ELEMENT>
  <ROUTES>
    <RE_IP1>0.0.0.0</RE_IP1>
    <RE_SM1>255.255.254.0</RE_SM1>
  </ROUTES>
  <DSP>
    <ECHO_CANCEL>1</ECHO_CANCEL>
    <DSP_ECHO_TAIL>128</DSP_ECHO_TAIL>
    <VAD_THRESHOLD>-17</VAD_THRESHOLD>
    <IDLE_NOISE>-65</IDLE_NOISE>
    <DTMF_TONE_DETECT>1</DTMF_TONE_DETECT>
```

```
<MODEM_DETECT>1</MODEM_DETECT>
<FAX_DETECT>1</FAX_DETECT>
<FAX_RATE>14400</FAX_RATE>
<FAX_PLAYOUT_NOM_D>100</FAX_PLAYOUT_NOM_D>
<FAX_ACTIVE_TIMEOUT>20</FAX_ACTIVE_TIMEOUT>
<FAX_PACKET_SIZE>30</FAX_PACKET_SIZE>
</DSP>
<CODEC>
  <CODEC_NUMBER>1</CODEC_NUMBER>
  <VX_PAYLOAD>10</VX_PAYLOAD>
  <VX_PLAYOUT_NOM_D>40</VX_PLAYOUT_NOM_D>
  <VX_PLAYOUT_MAX_D>80</VX_PLAYOUT_MAX_D>
  <VAD_ENABLED>0</VAD_ENABLED>
</CODEC>
<CODEC>
  <CODEC_NUMBER>5</CODEC_NUMBER>
  <VX_PAYLOAD>20</VX_PAYLOAD>
  <VX_PLAYOUT_NOM_D>40</VX_PLAYOUT_NOM_D>
  <VX_PLAYOUT_MAX_D>80</VX_PLAYOUT_MAX_D>
  <VAD_ENABLED>0</VAD_ENABLED>
</CODEC>
<CODEC>
  <CODEC_NUMBER>7</CODEC_NUMBER>
  <VX_PAYLOAD>30</VX_PAYLOAD>
  <VX_PLAYOUT_NOM_D>60</VX_PLAYOUT_NOM_D>
  <VX_PLAYOUT_MAX_D>120</VX_PLAYOUT_MAX_D>
  <VAD_ENABLED>1</VAD_ENABLED>
</CODEC>
<CODEC>
  <CODEC_NUMBER>2</CODEC_NUMBER>
  <VX_PAYLOAD>30</VX_PAYLOAD>
  <VX_PLAYOUT_NOM_D>60</VX_PLAYOUT_NOM_D>
  <VX_PLAYOUT_MAX_D>120</VX_PLAYOUT_MAX_D>
  <VAD_ENABLED>0</VAD_ENABLED>
</CODEC>
<CODEC>
  <CODEC_NUMBER>9</CODEC_NUMBER>
  <VX_PAYLOAD>30</VX_PAYLOAD>
  <VX_PLAYOUT_NOM_D>60</VX_PLAYOUT_NOM_D>
  <VX_PLAYOUT_MAX_D>120</VX_PLAYOUT_MAX_D>
  <VAD_ENABLED>0</VAD_ENABLED>
</CODEC>
<CODEC>
```

```

<CODEC_NUMBER>8</CODEC_NUMBER>
<VX_PAYLOAD>1</VX_PAYLOAD>
<VX_PLAYOUT_NOM_D>60</VX_PLAYOUT_NOM_D>
<VX_PLAYOUT_MAX_D>120</VX_PLAYOUT_MAX_D>
<VAD_ENABLED>0</VAD_ENABLED>
</CODEC>
<DIFFSERV>
<CONTROL_Prio>160</CONTROL_Prio>
<VOICE_Prio>184</VOICE_Prio>
<IEEE_802_1PQ_ENABLED>1</IEEE_802_1PQ_ENABLED>
<IEEE_802_1P>6</IEEE_802_1P>
<IEEE_802_1Q>0</IEEE_802_1Q>
<NAT_ENABLED>0</NAT_ENABLED>
<NAT_TIMEOUT>90</NAT_TIMEOUT>
</DIFFSERV>
<ELAN>
<CALL_SERVER_IP>47.11.216.167</CALL_SERVER_IP>
<SURVIVAL_IP>0.0.0.0</SURVIVAL_IP>
<SIGNAL_PORT>15000</SIGNAL_PORT>
<BROADCAST_PORT>15001</BROADCAST_PORT>
</ELAN>
<TLAN>
<SIGNAL_PORT>5000</SIGNAL_PORT>
<AUDIO_PORT>5200</AUDIO_PORT>
</TLAN>
<LOSS_PLAN>
<COUNTRY>UK</COUNTRY>
</LOSS_PLAN>
<GATE_KEEPER>
<PR_GK_IP>0.0.0.0</PR_GK_IP>
<ALT_GK_IP>0.0.0.0</ALT_GK_IP>
<PR_NCS_IP>47.11.215.126</PR_NCS_IP>
<PR_NCS_PORT>5000</PR_NCS_PORT>
<ALT_NCS_IP>0.0.0.0</ALT_NCS_IP>
<ALT_NCS_PORT>5000</ALT_NCS_PORT>
<NCS_TIMEOUT>10</NCS_TIMEOUT>
</GATE_KEEPER>
<FIRMWARE>
<FW_SERVER_IP>47.11.216.194</FW_SERVER_IP>
<FW_SERVER_SM>255.255.254.0</FW_SERVER_SM>
<FW_FILE_DIR_PATH>/u/fw</FW_FILE_DIR_PATH>
<USER_ID>admin1</USER_ID>
<PASSWORD>0000 </PASSWORD>

```

```

</FIRMWARE>
<APPSERVER_1>
  <IP>47.11.216.194</IP>
  <HOSTNAME></HOSTNAME>
  <H323ID>SS_1</H323ID>
  <SW_VTRK_TPS>1</SW_VTRK_TPS>
  <SW_GK>0</SW_GK>
  <SW_SET_TPS>1</SW_SET_TPS>
</APPSERVER_1>
<SNTP_SERVER>
  <MODE>active</MODE>
  <INTERVAL>256</INTERVAL>
  <PORT>20222</PORT>
</SNTP_SERVER>
<SNTP_CLIENT>
  <MODE>passive</MODE>
  <INTERVAL>256</INTERVAL>
  <PORT>20222</PORT>
  <SNTP_SERVER_IP>0.0.0.0</SNTP_SERVER_IP>
</SNTP_CLIENT>
</config>value = 0 = 0x0
-> httpDebugFlagxfiles=0

```

- **httpDebugFlagOvly = 2**

`/webxml/xmltmp/ovlinfo.xml` and `/webxml/xmltmp/input.xml` contain the last overlay CGI information received from a web browser if the flag is set. The input XML file is sent to CS. See the following example.

```

-> httpDebugFlagOvly=2
httpDebugFlagOvly = 0x2674130: value = 2 = 0x2
-> cd "/webxml/xmltmp"
value = 0 = 0x0
-> ll

```

size	date	time	name
512	OCT-16-2002	16:13:22	.
512	OCT-16-2002	16:13:22	..
48	OCT-16-2002	16:23:50	ovlinfo.xml
147	OCT-16-2002	16:23:54	input.xml

```
value = 0 = 0x0
-> copy < input.xml
<?xml version="1.0"?>
<LDOVL>
<LD>22</LD>
<LOGIN_NAME>henry</LOGIN_NAME>
<TIMESTAMP>301768788</TIMESTAMP>
<REQ>PRT</REQ>
<TYPE>PKG</TYPE>
</LDOVL>
value = 0 = 0x0
-> httpDebugFlagOvly=0
httpDebugFlagOvly = 0x2674130: value = 0 = 0x0
```

Administrator logout of Web Server users

Users may occasionally encounter a browser crash. When this occurs, the Web Server does not receive a logout message from the browser. If a user re-launches the browser and tries to login again, the login attempt fails because the Web Server still identifies the user as being logged in. To change the state of the user to logged out, the administrator must take the following steps:

- 1 Enter **userShow** in the vxshell of the Signaling Server.
The system displays a list of the users currently logged in.
- 2 Find the user experiencing the problem in the list.
- 3 Enter the **userLogOut** command to log the specified user out of the system.

The user can launch the browser and log in.

See the following example.

```
-> userShow
User Check task tHTTPUsr is up
+---User---+ActiveTime+Login Time+ Sequence +---Browser
IP---+Brows+OVL+-----CS IP-----+Langu+Flag+F+Buf L+
+ henry+1034785431+0x11fca054+0x00000001+ 47.10.34.60+MS IE+YES+
47.11.216.167+ 0 en+ OK ++ 0+
```

```
+ admin2+1034785489+0x14f790d2+0x00000002+ 47.10.34.60+MS IE+NO+
47.11.216.150+ 0 en+ OK ++ 0+
Total Entries 2
Total User Logged in 2
value = 23 = 0x17
```

```
-> userLogOut "admin2"
value = 0 = 0x0
```

```
-> userShow
User Check task tHTTPUsr is up
+---User---+ActiveTime+Login Time+ Sequence +---Browser
IP---+Brows+OVL+-----CS IP-----+Langu+Flag+F+Buf L+
+ henry+1034785431+0x11fca054+0x00000001+ 47.10.34.60+MS IE+YES+
47.11.216.167+ 0 en+ OK ++ 0+
Total Entries 1
Total User Logged in 1
value = 23 = 0x17
```

Remote Procedure Call

The RPC interface interfaces between the Web Server and the Call Server, Signaling Server, or Voice Gateway Media Card.

RPC limitations

The limitations of the RPC component are:

- Buffer size—An RPC message cannot exceed 96kb. If the limit is exceeded, the following messages will appear. If this occurs, you must re-issue the command.
 - **WEB3101**—Input XML file overflow.
 - **WEB3102**—Output XML file overflow.
 - **WEB4001**—XML output buffer overflowed.
- File size—**Bootp.tab** has a maximum file size of of 4kb while the **config.ini** file has a maximum file size of 8kb. If either of the files is larger than the maximum file size and you submit a node, the following errors are displayed:

- **WEB3203**—New configuration file is invalid.
- **WEB3204**—New bootp file is invalid.

RPC troubleshooting tips

Troubleshooting the RPC Server on the SS or VGMC

The RPC server runs on the VGMC, the SS, and the CS. The RPC client runs on the Web Server, which resides on the SS. For every message received from the RPC client, the RPC module on the VGMC and the SS generates a syslog information message for output. These messages can be monitored through a tip or telnet session to the card.

The syslog information messages indicate whether the RPC module has successfully received the messages from the Web Server.

The following example shows the syslog messages generated on the VGMC or SS as a result of issuing the **i**, **ifShow** and **ping** commands on the IP telephony status page.

```
[0070] 07/08/02 13:02:22 LOG0006 tRPCMGMT: Got general command request
from RPC client: i
[0071] 07/08/02 13:02:27 LOG0006 tRPCMGMT: Got general command re-
quest from RPC client: ifShow
[0073] 07/08/02 13:02:50 LOG0006 tRPCMGMT: Got general command request
from RPC client: ping 47.11.216.167 3
```

Troubleshooting the RPC Server on the CS

You can set a debug flag for the RPC server running on the Call Server to aid in troubleshooting. The commands associated with setting and clearing the flag must be issued from the PDT shell on the CS. The commands for setting and clearing the flag are:

- **setRpcDebug**—sets the debug flag, enables debugging
- **clearRpcDebug**—clears the debug flag

Note: All messages will be displayed on tip connections only, rlogin sessions do NOT display any messages.

The following example illustrates the results of a trace on the CS during a user login to EM.

```
pdt> symload
pdt> setRpcDebug
value = 1 = 0x1

pdt> RPC got input XML file from client side is:
<?xml version="1.0"?>
<VALIDATION>
<LOGIN_NAME>henry</LOGIN_NAME>
<TIMESTAMP>883333298</TIMESTAMP>
<PASSWORD>0000</PASSWORD>
</VALIDATION>
RPC server got correct return from Transaction server.
RPC server got output XML file from Trx Server is:
<?xml version="1.0"?>
<VALIDATION>
<RESULT>VALID</RESULT>
</VALIDATION>

pdt> clearRpcDebug
value = 1 = 0x1
```

IP Telephony node configuration tips on the Call Server

- When node data is imported by the Call Server from the Signaling Server using Element Manager, the files are updated whenever the node is used.
- All user configured node information is registered or deregistered in the file, **node.pch**, which is a binary file. To display the current registered node, enter the command **printNodeList** in the PDT shell on the CS.
- The **bootp.tab** and **config.ini** files are saved as **c:/u/db/node/nodexxxx.btp** and **c:/u/db/node/nodexxxx.cfg** where **xxxx** is the node ID.
- All the registered node files and the **node.pch** file are copied to the **z:/** drive on the Call Server when you backup the CS.

- When you perform a restore, the Call Server backs up the current node files in **c:/u/db/node** by renaming them and then restores the files from copies on the **z:/** drive. You can recover the old version of the files by copying the files. The backup file naming convention is:
 - **node.pch->node.bak**
 - **nodexxxx.btp->nodexxxx.bbt**
 - **nodexxxx.cfg->nodexxxx.bcf**

Note: These backup files will be deleted when you delete the node.



CAUTION — Service Interruption

Bootp.tab and **config.ini** files have a special format. Editing or deleting these files manually may cause EM corruption.

Transaction Server

The Transaction Server provides an interface between RPC and the Call Server. The Transaction Server also has two debug flags, which can be set from the Call Server. If the flags are set, a table is displayed, containing information based on the input XML file just before being sent to the overlays.

After you have collected the transaction debug information, it is strongly recommended that you deactivate the debug flags before the next transaction.

To set the flags, you must have access to the PDT shell on the CS. Before setting the flags, ensure that symbols are loaded. The two Transaction Server debug flags are **TrxServerDebug** and **SMP_nScriptRunDebug**.

TrxServerDebug

To set this debug flag, do the following:

- 1 Look up the symbol.

```
pdt> symload
pdt> lkup "TrxServerDebug"
TrxServerDebug__mainTrx_cxx 0x203a79d8 data
```

- 2 Write '1' to the address. Enter a "." when prompted with the next address. This will signify the end of the change.

```

pdt> m 0x203a79d8
203a79d8: 0000-1<CR>
203a79da: 0000-.<CR>

```

To clear the flag, write '0' to the address. Enter a "." when prompted with the next address. This will signify the end of the change.

```

pdt> m 0x203a79d8
203a79d8: 0000-0<CR>
203a79da: 0000-.<CR>

```

Note 1: All the messages will only be displayed on tip connections, rlogin session will NOT show any message.

Note 2: The address of the flag may change between software versions. Always look up the flag address before attempting to set or clear the flag. Writing to the wrong address can cause problems on the CS.

The following example illustrates the results of a debug trace, captured on the CS side, while a user displays the time on CS using overlay 2.

```

*****
Start_Trx function start
*****
Start to create table
This is header:LDOVL
LD      2
LOGIN_NAME      henry
TIMESTAMP      93333296
COMMAND      TTAD
Start to get session!

The channel we got is:6
start to load overlay
Prompt search is OK for LOGIN_NAME -> henry

```

```
Prompt search is OK for LD -> 2
-- Inbuffcnt = 0 < 800
Got command: COMMAND
Prompt search is OK for COMMAND -> TTAD
Write to switch now, value is: TTAD.
-- Inbuffcnt = 4 < 800
Got command: COMMAND
Transaction is successful, Trx Server frees channel:6
XML output buffer size: 182
*****
<?xml version="1.0"?>
<LDOVL>
<DATE>
<DAY>16</DAY>
<MON>10</MON>
<YEAR>2002</YEAR>
</DATE>
<TIME>
<HOUR>10</HOUR>
<MIN>23</MIN>
<SEC>16</SEC>
</TIME>
<RESULT>VALID</RESULT>
</LDOVL>
```

SMP_nScriptRunDebug

To set this debug flag, do the following:

- 1 Look up the symbol.

```
pdt> symload
pdt> lkup "SMP_nScriptRunDebug"
_SMP_nScriptRunDebug 0x203af8c4 data
```

- 2 Write '1' to the address. Enter a "." when prompted with the next address. This will signify the end of the change.

```
pdt> m 0x203af8c4
203af8c4: 0000-1<CR>
203af8c6: 0000-.<CR>
```

To clear the flag, write '0' to the address. Enter a "." when prompted with the next address. This will signify the end of the change.

```
pdt> m 0x203af8c4
203af8c4: 0001-0<CR>
203af8c6: 0000-.<CR>
```

The following example illustrates the results of a debug trace captured while a user was logging into EM.

```
pdt> lkup "SMP_nScriptRunDebug"
_SMP_nScriptRunDebug 0x203af8c4 data
pdt> m 0x203af8c4
203af8c4: 0000-1<CR>
203af8c6: 0000-.<CR>

value = 1 = 0x1
pdt> getSession():
getNextPty():
Free port 1
phys tty = 6, log tty = 5, master fd = 66, slave fd = 67
loadOverlay(0, 5,[henry:0000],0): XML_FLAG = 1
username & password ok
freeSession(5)
phys tty = 6

pdt> m 0x203af8c4
203af8c4: 0001-0<CR>
203af8c6: 0000-.<CR>

value = 1 = 0x1
```

Data networking

Call Server and Media Gateway (SIPE) diagnostics

This section describes some of the CLI commands you can use to view the status and configuration of survivability, and the Call Server to Media Gateway IP links.

Note: The overlay 117 commands contained in this section apply to small systems (SSC) only.

Overlay 135 STAT IPL <cab>

Display the IP link status between the Call Server and the MG 1000S.

Overlay 135 STAT GR

Display the current status of Geographic Redundancy.

Overlay 117 PRT IPM <port>

Print the IP connectivity configuration data associated with the Main Cabinet end of the specified port.

Overlay 117 PRT IPR <port>

Print the IP connectivity configuration data associated with the Expansion Cabinet end of the specified port.

Overlay 117 STAT AUTONEG IPM

Display the auto-negotiate status of the Main Cabinet ports.

**AUTO-NEGOTIATE LINK PARTNER STATUS -
MAIN/CALL SERVER PORTS**

PORT Bandwidth Duplex Mode AutoNegotiate
=====

**IPR 1 UNKNOWN UNKNOWN ON
IPR 2 UNKNOWN UNKNOWN
IPR 3 100 Mbps full duplex ON
IPR 4 UNKNOWN UNKNOWN**

If the auto-negotiation process is successful, the message, "**100 Mbps full duplex**" is displayed. If the auto-negotiation process is not successful, the message "**UNKNOWN**" is displayed, indicating a failure in negotiating 100 Mbytes per second full duplex bandwidth.

Overlay 117 STAT AUTONEG IPR

Display the auto-negotiate status of the expansion cabinet ports.

**AUTO-NEGOTIATE LINK PARTNER STATUS -
EXPANSION/MEDIA GATEWAY PORTS**

PORT Bandwidth Duplex Mode AutoNegotiate
=====

**IPR 1 UNKNOWN UNKNOWN ON
IPR 2 UNKNOWN UNKNOWN
IPR 3 100 Mbps full duplex ON
IPR 4 UNKNOWN UNKNOWN**

If the auto-negotiation process is successful, the message, "**100 Mbps full duplex**" is displayed. If the auto-negotiation process is not successful, the message, "**UNKNOWN**" is displayed, indicating a failure in negotiating 100 Mbytes per second full duplex bandwidth.

Overlay 117 PRT SURV <cab>

Print the Expansion Cabinet Survivable capability for all, or for the specified Expansion Cabinets.

Overlay 117 PRT CAB <cab>

Print the parameters and survivable capability of the specified Expansion Cabinet.

PDT shell and LDB shell

- PDT is the VxWorks shell on the Call Server.
- LDB is the VxWorks shell on the Media Gateway unless the Media Gateway is in survival mode. PDT is the VxWorks shell on the Media Gateway if the Media Gateway is in survival mode.

ELAN troubleshooting tips

- CS Media Gateway 100BaseT links must be point-to-point, L2, or L3. There must be absolutely no hubs as collisions may occur!!
- Watch for "**SRPT1027 STARTUP: IP port is not 100BaseT Full Duplex**" messages. If the cabinet has registered, this message indicates a link problem that must be fixed, otherwise call process problems will occur!
- Check the status of the IP daughterboard LEDs on the circuit board, near the RJ-45 connectors (not the LEDs on the faceplate). The normal operating conditions of the LEDs are:
 - link LED (green)—solid,
 - tx LED (yellow)—flashing, or almost solid,
 - rx LED, (red)—flashing, or almost solid.

IP Line 4.0 Troubleshooting

Information about troubleshooting IP Line 4.0 and the Voice Gateway Media Card (VGMC) is available in the *IP Line Release 4.0 Troubleshooting Guide for Distributors* document.

Nortel Communication Server 1000

CS 1000 Release 4.0

Troubleshooting Guide for Distributors

Expert Guide

Copyright © 2006 Nortel Networks. All Rights Reserved.

The information in this document is subject to change without notice. The statements, configurations, technical data, and recommendations in this document are believed to be accurate and reliable, but are presented without express or implied warranty. Users must take full responsibility for their applications of any products specified in this document. The information in this document is proprietary to Nortel Networks.

Nortel, Nortel (Logo), the Globemark, SL-1, Meridian 1, and Succession are trademarks of Nortel Networks.

Document release: Standard 2.00

Date: February 2006

Produced in Canada

