



Touchstone Release AR01.1

Firmware Guide

STANDARD Revision 1.0

January 2018

© 2018 ARRIS Enterprises LLC. All Rights Reserved.

Touchstone® AR01.1 Firmware Guide

STANDARD 1.0

ARRIS Copyrights and Trademarks

© 2018 ARRIS Enterprises LLC. All Rights Reserved.

No part of this publication may be reproduced in any form or by any means or used to make any derivative work (such as translation, transformation, or adaptation) without written permission from ARRIS Enterprises LLC. ("ARRIS"). ARRIS reserves the right to revise this publication and to make changes in content from time to time without obligation on the part of ARRIS to provide notification of such revision or change.

ARRIS and the ARRIS logo are all trademarks of ARRIS Enterprises LLC. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks or the names of their products. ARRIS disclaims proprietary interest in the marks and names of others.

ARRIS provides this guide without warranty of any kind, implied or expressed, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. ARRIS may make improvements or changes in the product(s) described in this manual at any time.

The capabilities, system requirements and/or compatibility with third-party products described herein are subject to change without notice.

Table of Contents

1. Overview	14
About This Manual.....	14
Supported Hardware	15
Firmware Functionality.....	15
Standards Compliance	16
Standard Functionality.....	16
Optional Functionality	16
DOCSIS Specifications	16
CableLabs IPv6 Specifications	17
DOCSIS 3.1 Security Overview	17
Load Name Extensions.....	17
2. DOCSIS Provisioning	19
General Provisioning Information	19
Service Flow Limitations	19
OUI Ranges.....	19
Configuring ToD Offset	19
Setting the DST Policy	19
Clearing the CPE List	20
Provisioning Considerations	20
Interface Index Scheme	20
Configuration File Provisioning Notes	22
Support for TLV-41 (Downstream Channel Lists)	22
Persistence.....	22
Cable Modem Interface Mask.....	23
CM DHCP Interactions	23
DHCP Option 51 Support	24
DHCP Option 60 Support	24
Dual-Mode Operation.....	24
Forcing Provisioning Mode and Certificates	24
Configuration Files and Signed Loads	25
Certificates	25
DMDM Status.....	25
Overriding the MDD IPv4/IPv6 Selection	25
Displaying the MDD Setting	26
Configuring Extended Upstream Transmit Power.....	26
Upgrading Touchstone Firmware	26

Action	26
Upgrading the Firmware through Provisioning	27
Upgrading the Firmware through SNMP	27
Configuring Channel Bonding Characteristics	28
Action	28
IPv6 Provisioning Notes	28
IPv6 Provisioning Modes.....	28
Selecting an Addressing Mode.....	28
DHCP Behavior for IPv6 Provisioning.....	29
Provisioning File Notes.....	30
TLV-38 Enhancements	30
Configuring the Diplexer.....	30
3. NCS Voice Provisioning.....	32
NCS Provisioning Considerations.....	32
About IPsec	32
Call Management Servers	32
Voice and Signaling Ports.....	32
eDVA Interface Table	33
Provisioning Modes	33
PacketCable Provisioning Modes.....	33
Verifying eDVA Provisioning and Endpoint Status.....	36
DHCP Support by Provisioning Mode	36
Options Required in All Provisioning Modes	36
PacketCable Modes	37
eDVA DHCP Interactions.....	38
DHCP Option 43 Support	38
DHCP Option 51 Support	38
Disabling Option 122 Sub-Option 3 Enforcement	39
Provisioning Quality of Service	39
Full DQoS Mode	39
DSX QoS Mode	39
Feature Switches	40
CallP Feature Switch	40
Secondary CallP Feature Switch.....	48
Provisioning General eDVA and Line Parameters	49
Action	49
Setting Persistent Line Status	49
Controlling ToS Byte Marking	49

Controlling TurboDOX Functionality.....	50
Controlling IPsec Functionality	50
Configuring the Ringing Waveform	50
Configuring Loop Current	51
Configuring Caller ID Options	52
Setting Loop Voltage Management	52
Loop Voltage Management Policies	53
Loop Voltage Management MIB Objects.....	54
Action.....	56
Echo Cancellation and Analog Fax/Modem Support.....	56
Adaptive Jitter Buffers	57
Configuring the Echo Cancellation Tail Length	57
Provisioning RFC 2833 Support	57
Controlling RFC 2833 Functionality.....	58
Configuring T.38 Fax Relay Support.....	58
SDP Parameter List for T.38 Strict.....	58
PacketCable 1.5 Extended Signaling	63
SDP Parameter List Considerations	64
T.38 Provisioning Overview	64
Action	65
Controlling T.38 and Fax-Only Modes	65
Configuring T.38 MaxDatagram Size.....	66
Super G3 FAX Support	66
Configuring Jitter Buffers.....	67
Action	68
Setting Standard Jitter Buffer Parameters.....	68
Setting Voice Band Data Jitter Buffer Parameters.....	68
Configuring Custom Jitter Buffer Settings	69
Configuring Call Progress Tones	70
MIB Tables	70
Action	71
Gain Compensated Tone Generation	72
On-Hook vs. Off-Hook Gain.....	72
Action.....	72
Configuring Gain Control using SNMP	72
Provisioning Preset Downstream Frequencies.....	74
Preset Frequency MIB Objects	74
Dial Pulse Support.....	75
Inband DTMF Transmission	75

Action	76
Gateway Dial Pulse Example	76
Configuring Hook Flash Timing	77
Default Timing Settings	77
Action	77
Provisioning Ring Cadences	77
Post-Provisioning	78
NCS Post-Provisioning	78
Action	78
4. Provisioning ARRIS SIP Loads	79
Overview of SIP Features	79
Barge-In	79
Loopback	79
Extended Offhook Processing	79
Emergency Calls	81
Distinctive Ringing	82
SIP Provisioning Considerations	82
Information Required for SIP	82
SIP Registration Behavior	84
SIP Feature Switch	84
Provisioning Details	85
Minimal Example	91
Provisioning SIP Support	91
Per-Line Proxy/Registrar Objects	91
T.38 Provisioning Overview	92
Global Call Feature Control	93
Per-line Call Feature Control	94
Action	95
CM Configuration File Changes	95
eDVA Configuration File Changes	95
Setting up Timers	97
Configuring Per-Line Proxy and Registrar	98
Specifying a SIP Domain Name	99
Provisioning SIP Features	99
Requirements and Limitations	99
Call Feature Control	99
Proxy Dialing Features	100
Supported Dialing Features	101

Action	102
Setting up Dialing Features	102
Configuring Warmline or Hotline.....	103
Configuring Repeat Dialing	104
Configuring T.38 and Fax-Only Modes.....	105
Configuring Distinctive Ring/Alert Tones.....	105
5. Provisioning PacketCable 2.0 SIP Loads.....	107
PacketCable 2.0 Concepts	107
Terminology	107
Configuration Concepts	108
Supported Features	109
DHCP Option 60 Support	110
Overview of SIP Features.....	111
Barge-In.....	111
Loopback.....	111
Extended Offhook Processing.....	112
Emergency Calls	113
Distinctive Ringing.....	114
Configuring PacketCable 2.0 SIP	114
Configuration Overview	114
Action	115
Configuring Operator Information.....	115
Configuring Users and Features.....	116
Configuring Extended Offhook Processing	116
Post-Provisioning SIP Lines	117
Configuring T.38 and Fax-Only Modes.....	118
Provisioning PacketCable 2.0 Features.....	118
Feature Support	118
P-CSCF Dialing Features	120
Action	121
Basic Call Configuration	121
Configuring the Status Change Feature	122
Configuring No Answer Timeout	122
Configuring Caller ID	122
Configuring Emergency Services.....	123
Configuring Distinctive Ring/Alert Tones.....	124
Configuring PacketCable 2.0 Digit Maps	124
General Digit Map Structure.....	124

AR01.1 Compliance with PacketCable 2.0	127
Specifying a Digit Map	127
Example Digit Map	127
Provisioning PacketCable 2.0 Users.....	130
Action	130
Provisioning PacketCable 2.0 Application Profiles	132
Indexing.....	132
Action	132
Provisioning PacketCable 2.0 Application Maps	133
Prerequisites	133
Indexing.....	133
Action	133
Configuring SIP Failure Response Tones.....	134
Priority	134
Action.....	134
Playing Busy Tone for All Errors	134
Configuring Individual Response Tones.....	135
Configuring MWI Support.....	135
Action	135
Clearing MWI Indicators	136
Provisioning the MWI Subscription	136
Voice Mail Subscription Watchdog.....	136
6. Provisioning a Gateway (eRouter)	137
Gateway Provisioning Methods.....	137
Provisioning Precedence.....	137
eRouter Operating Modes	138
eRouter Wi-Fi Country Codes	138
LAN-side Devices.....	138
Routed Network Devices	139
Default eRouter Settings.....	139
MoCA Configuration Notes.....	139
eRouter IPv6 Operation	140
TR-069 Provisioning	140
Enabling TR-069 Support	140
TR69AcInfo Sub-TLV Formal Definitions	140
Obtaining TR-181 Parameters Using DHCP.....	142
TLV-202 Based Provisioning.....	143
TR69ManagementServer Sub-TLV Definitions	144

Gateway DHCP Interactions	145
WAN Interface Dynamic Provisioning.....	145
IPv6 eRouter Considerations	146
Setting the Operating Mode.....	146
Configuring the Wireless Channel	147
Configuring DNS Override/Relay	147
Configuring IPv6 DHCP Services.....	148
Managing Network Extenders (AR01.1)	148
Managing Network Extenders from the Gateway.....	148
Managing Network Extenders Using SNMP.....	149
Provisioning Home Hotspot.....	149
Supported Hardware	149
Home Hotspot Functionality.....	149
Action.....	150
Guest SSID.....	151
Band Steering.....	151
Band Steering Requirements	152
Configuring Band Steering	152
ARRIS-ROUTER-MIB Objects Supported in AR01.1.....	153
Mapping ARRIS-ROUTER-MIB Objects to RDK-MIB Objects.....	154
7. Operations.....	157
Battery Management	157
Initial Battery Charging	157
Battery Telemetry	157
Power Failure Operation.....	157
LED Changes.....	158
Battery Status Monitoring	158
Highest Charger Temperature Recording.....	158
Advanced Power Management	158
CMTS Considerations.....	159
About IPv6 Support.....	159
Supported Hardware	160
IPv6 Multicast Support.....	160
IPv6 Management.....	161
Filtering IPv6 Traffic	161
Coexistence.....	162
DHCPv6 MIB Objects.....	163
SNMP Access.....	164

Event Reporting	164
Collecting Events.....	165
Event Formats	165
References	166
Event Summary.....	167
Event Handling	167
eDVA States.....	167
eDVA Line States	168
E-UE Battery States	168
eDVA Event Summary	169
ARRIS Events	171
Voice Line Diag Failed	171
Voice Line Diag Passed.....	172
Voice Line State Change.....	172
Voice Line Protection State Change.....	172
Power Supply Telemetry Log	173
MTA TFTP: Successful	173
MTA PROV: Successful!.....	173
SSH LOGIN ACCEPTED	174
SSH LOGIN REJECTED	174
SSH LOGIN REJECTED - MAX ATTEMPTS REACHED.....	174
Touchstone Firmware Upgrade Failed Before Download Attempt.....	175
Touchstone Firmware Upgrade Failed.....	176
Touchstone Firmware Upgrade Successful.....	176
Touchstone SW Upgrade Aborted due to Battery AC-FAIL condition	176
Touchstone SW Upgrade Aborted due to Call in Progress	177
Touchstone SW Upgrade Reboot Delayed due to Call in Progress.....	177
MTA DHCP RENEW: Lease Renewal delay; Voice line offhook.....	177
MTA DHCP REBIND: Lease Renewal delay; Voice line offhook.....	178
Power Supply Telemetry Alarm	178
Gateway has reset	178
Unit has been restored to factory defaults	179
Voice Line Provisioning Complete	179
State Changed	179
MTA TFTP: Failed	180
Loop Voltage Management: Policy Missing.....	180
Loop Voltage Management: Bad Key	180
Loop Voltage Management: Policy Out of Range.....	180
Loop Voltage Management: Policy Change	181

Loop Voltage Management: Policy 3 Timer Out of Range	181
Call Stats.....	181
Last NCS Message Received.....	182
Power Supply Telemetry Alarm - Battery Missing.....	182
Power Supply Telemetry Alarm - Battery Low.....	183
Power Supply Telemetry Alarm - Replace Battery.....	183
Voice Line Failure	183
PacketCable Events.....	184
Battery Not Low	184
Battery Low	184
Battery Present	184
Battery Missing	184
Battery Good.....	184
Replace Battery.....	184
AC Restored	184
AC Fail	185
Network Failure Recovery	185
Recovery from Extreme Plant Conditions.....	185
Working with Message Trace Logs	186
Message Capacity	186
SNMP Overview	186
Action.....	186
Enabling or Disabling Message Tracing.....	187
Viewing Logs Using SNMP.....	187
Capturing Signaling Traces.....	188
Controlling Signaling Tracing	188
Interpreting the Signaling Trace Output Data	189
Configuring SNMP Coexistence	192
Overview	192
snmpCommunityTable Parameters	194
vacmSecurityToGroupTable Parameters	195
vacmAccessTable Parameters	196
Action.....	196
Adding the snmpCommunityTable	197
Adding the vacmSecurityToGroupTable	199
Adding the vacmAccessTable.....	201
Configuring Trap Servers	204
Action.....	204
Power Management	207

Recovery from Partial Service.....	207
Action.....	208
Identifying Partial Service Issues	208
If the CMTS does not Support CM-STATUS.....	208
If the CMTS does not Support REG-ACK	209
ARRIS DOCSIS 3.0 MIB	209
arrisCmDoc30Base	209
arrisCmDoc30Access.....	210
arrisCmDoc30Setup	210
arrisCmDoc30Dhcp	213
arrisCmDoc30DhcpExtended	214
arrisCmDoc30ResetReasonLog	217
HD Audio MIB Objects	217
DOCSIS 3.0 MIB Object Mapping	218
Supported eDVA MIB Objects	219
PACKETPORT-MIB Objects	219
ARRIS-MTA-MIB (non-battery)	220
ARRIS-MTA-MIB (battery telemetry items)	237
8. Administration.....	242
Administration Objects.....	242
System Description Objects	242
Bridging and Routing Objects	243
End of Call Connection Statistics	244
NCS Behavior.....	245
SIP Behavior	245
End-of-Call Statistics MIB Objects.....	246
Clearing Counters.....	248
Last Signaling Message Sent	249
Per-Call Syslog Reporting	249
Using the Speedtest Application	252
Server Requirements	252
Running a Speed Test using TR-143 Objects.....	252
Network Performance Monitoring	254
Test Types	254
Setup	254
Configuring Tests	255
Running Tests.....	256
Results.....	257

9. Maintenance	260
Overview of Maintenance Interfaces	260
WebGUI Access Levels and Defaults	260
LED Patterns	260
Wiring Problems Indication	260
TM3402 Normal Operation	261
Loopback Testing	263
Reset to Factory Defaults	263
Using the Password of the Day Tool	263
About the Password of the Day Tool	263
Action	264
Changing the Seed	264
Generating a Single Password	265
Generating a List of Passwords	265
Using the Spectrum Analyzer	265
10. References	267
Supported Calling Features	267
Country Code Templates	268
North American Ring Cadences	270
Customizing Default Ring Cadences	271
Default Tone Settings	271
North America	272
CableLabs Wi-Fi Objects MIB	273
MoCA MIB	287
mocalfConfigTable	287
mocalfStatusTable	289
hneMIB Objects	291
hneWiFiGWSupport Objects	292
TR-069 Management	295
Overview	295
Supported TR-181 Objects	297
References	376

Overview

Touchstone devices provide the subscriber connection to the HFC IP network.

Touchstone devices running AR01.1 firmware comply to the following standards:

- DOCSIS 3.1
- PacketCable 1.5 and PacketCable 2.0

About This Manual

This manual describes Touchstone® AR01.1 firmware.

Some features described in this manual may not be fully tested and supported in your specific firmware release version. Where possible, features supported only by specific versions are indicated in this manual. See the *Release Notes/Letter of Operational Considerations* accompanying your firmware for further details.

Audience

This manual assumes that you have a basic understanding of DOCSIS and PacketCable standards, and a working knowledge of cable data and telephony networks.

In This Manual

This manual contains the following chapters:

- Chapter 1, "Overview," describes the Touchstone firmware and documentation, standards compliance, and load variants.
- Chapter 2, "DOCSIS Provisioning," describes provisioning the Cable Modem (CM) component of Touchstone devices.
- Chapter 3, "NCS Voice Provisioning," describes provisioning NCS telephony services on Touchstone devices.
- Chapter 4, "Provisioning ARRIS SIP Loads," describes features and services available for ARRIS SIP loads.
- Chapter 5, "Provisioning PacketCable 2.0 SIP Loads," describes special features and services available for SIP PC20 loads.
- Chapter 4, "Operations," describes monitoring, fault detection, and alerting mechanisms.
- Chapter 5, "Administration," describes performance statistics and maintaining system reliability.
- Chapter 6, "Maintenance," describes firmware updates, diagnostics, and troubleshooting features.

- Chapter 7, “References,” describes calling features, default ring cadences, and tones for each supported country code.

Supported Hardware

AR01.1 supports the following Touchstone models.

- DG3450
- TG3442, TG3452
- TG3492

Firmware Functionality

Touchstone AR01.1 firmware provides the following functionality:

- Supports Touchstone Model 34 products.
- Compatibility with DOCSIS 3.1.
- Supports mixed-mode provisioning (for example, DOCSIS 3.1 OFDM downstreams and DOCSIS 3.0 upstreams).
- Interoperability with ARRIS and other CMTS products.
- Supports up to 32 downstream DOCSIS 3.0 bonded channels and up to eight upstream bonded channels.
- Supports up to two 192 MHz OFDM bonded downstream channels and up to two OFDMA upstream channels.
- North American DOCSIS 3.1 loads support both 42/108 MHz and 85/108 MHz splits.
- Euro-DOCSIS 3.1 loads support 85/108 MHz and 204/258 MHz splits.
- Supports NCS, ARRIS SIP, and PC2.0 SIP telephony..
- Supports Ethernet interfaces to personal computers.
- Enhanced web-based troubleshooting interface.
- IPv4 and IPv6 addressing.
- Enhanced power management.

Firmware Download Center

ARRIS provides the ability to download firmware updates over the Internet, using the ARRIS Software/Firmware Delivery Tool. Benefits include an archive of released loads, and email notification of updated loads.

To obtain an account, contact ARRIS Technical Support.

High-level Changes from DOCSIS 3.0

DOCSIS 3.1 implements several significant improvements and changes.

- Hierarchical QoS
- Low Density Parity Code (LDPC) FEC enables use of higher-order modulation levels, including: 1024, 2048, 4096-QAM

- New downstream PHY:
 - New modulation scheme: Orthogonal Frequency Division Multiplexing (OFDM)
 - Many closely-spaced orthogonal subcarrier signals carry parallel data streams
 - Each sub-carrier uses a QAM modulation scheme at a low symbol rate
- New upstream PHY:
 - New modulation scheme: Orthogonal Frequency Division Multiple Access (OFDMA), a multi-user version of OFDM
 - Assigns groups of subcarriers to individual users
 - Allows simultaneous low data rate transmission from several users
- Larger PDU sizes
- CM reports downstream cable plant subcarrier mean error rate (MER)

To take advantage of DOCSIS 3.1 throughput enhancements, network upgrades may be needed. See the ARRIS white paper, *Preparing for DOCSIS 3.1: A Methodical Approach to Assessing Network Readiness* (<http://www.arris.com/globalassets/resources/white-papers/preparing-for-docsis-3-1.pdf>), for more information.

Standards Compliance

This section outlines Touchstone DOCSIS® and PacketCable™ compliance, and describes ARRIS-proprietary extensions to the standards.

Standard Functionality

Touchstone devices running version AR01.1 firmware comply with the following standards:

- DOCSIS 3.1
- PacketCable 1.5
- PacketCable 2.0

Optional Functionality

AR01.1 supports the following optional functionality specified by DOCSIS and PacketCable standards:

- Support for 10 ms and 20 ms packetization rates
- Support for up to 32 Upstream Service Flows
- Support for analog Fax/Modems, including automatic tone detection, echo cancellation disable and switching to the G.711 CODEC

DOCSIS Specifications

All DOCSIS specifications are available at the [DOCSIS web site](http://www.cablemodem.com/specifications/) (<http://www.cablemodem.com/specifications/>).

- *DOCSIS 3.0 Radio Frequency Interface Specification*, CM-SP-RF1v2.0-I11-060206

- *DOCSIS 3.1 Operations Support System Interface Specification*, CM-SP-CM-OSSiv3.1
- *DOCSIS 3.1 Security Specification*, CM-SP-SECv3.1
- *DOCSIS 3.1 Physical Layer Specification*, CM-SP-PHYv3.1
- *DOCSIS 3.1 MAC and Upper Layer Protocols Interface Specification*, CM-SP-MULPIv3.1
- *DOCSIS 1.1 Baseline Privacy Plus Interface Specification*, SP-BPI+-I12-050812
- *DOCSIS 1.1 Cable Modem to Customer Premise Equipment Interface Specification*, CM-SP-CMCiv3.0-I01-080320
- *Layer 2 Virtual Private Networks*, CM-SP-L2VPN-I09-100611

CableLabs IPv6 Specifications

- *CableLabs Assigned Names and Numbers*, CL-SP-CANN-I01-070119
- *CableLabs DHCP Options Registry*, CL-SP-CANN-DHCP-Reg-I01-070119

DOCSIS 3.1 Security Overview

AR01.1 supports a subset of the DOCSIS 3.1 Security Specification. In practical terms, this means:

- CLI access is *ssh* only, requires the Password of the Day for access, and highly restricted on production units. Debug devices (identified by an orange enclosure) use the *arristi* password and have full CLI access.
- All WebGUI access uses HTTPS, rather than unencrypted HTTP.
- Debug loads are no longer provided. Instead, separate debug devices provide debugging capabilities.

Load Name Extensions

The complete load name format is **NN.vv.vv.XXX_date_ss.cp.ll.special.i.dcs.sb**, where:

- **NN.vv.vv** is the firmware version:
 - TS11.01 for TM34xx models
 - AR01.01 for DG34xx and TG34xx models
- **XXX** is the revision number
- **date** indicates the date (in MMDDYY format) when the load was built

The following extensions are used to further identify hardware and signaling support in the load.

ss

The SDK release number. For example, **70** for release 7.0.

cp

Telephony signaling type supported:

- **NCS**
- **PC15** (PacketCable 1.5 NCS)

- **PC20** (PacketCable 2.0 SIP)
- **SIP** (ARRIS SIP)

ll

Load type. Currently supported load types are:

- **03**: TM3402
- **10**: DG34xx/TG34xx

special

Tags indicating special functionality (for example, **MAC14**).

i

A bitmask indicating the images included in the load:

- bit 0: ARM
- bit 1: Atom
- bit 2: UEFI
- bit 3: undefined

dcs

One of:

- **NA**: DOCSIS 3.0 load, signed with a North American certificate
- **EU**: DOCSIS 3.0 load, signed with a European certificate
- **D31**: DOCSIS 3.1 load

sb

Security indicator, one of:

- **.simg**: Secure Boot image
- **.img**: DOCSIS only signed image
- **.bin**: unsigned image

Example

The following load name:

AR01. 01. 123_121216_20. NCS. 10. 7. D31. si mg

is a DOCSIS 3.1 load for a DOCSIS 3.1 Telephony Gateway, supporting NCS telephony.

DOCSIS Provisioning

All Touchstone devices have a DOCSIS-compliant cable modem (CM) component. This chapter provides information for DOCSIS provisioning and ARRIS extensions.

General Provisioning Information

This section provides a general overview of provisioning-related information.

Service Flow Limitations

Touchstone devices support up to 32 service flows:

- Best Effort: 8
- UGS and UGS-AD: 24

OUI Ranges

AR01.1 uses 0000CA as the Vendor ID in DHCP messages. ARRIS issues periodic Field Bulletins with the most current listings.

Configuring ToD Offset

Time of Day changes, including for Daylight Savings Time shifts, normally occur during DHCP Renew operations. It may be necessary to manually change the offset, especially in SIP deployments where the eDVA clock is used as a source for Calling Line Presentation (CLIP) information.

1. To change the ToD offset, set the [arrisCmDoc30SetupTODTimeOffset](#) MIB object to the desired offset (in seconds). Valid range: **-43200** (-12 hours) to **46800** (+13 hours).



Note: This object is only accessible through the CM IP address.

When a DHCP exchange occurs (Renew, Rebind, or initial), the offset specified in the DHCP exchange overrides the value set in this MIB object.

Setting the DST Policy

DST policies may vary due to national legislation or local preferences. Touchstone firmware provides control over local DST policy.

1. To change the DST policy, set the **arrisCmDoc30SetupDSTPolicy** object to a string with the following format:

start=month/day/weekday/hour;end=month/day/weekday/hour

where...	is...
month	the month: 1 for January, to 12 for December.
day	The day: -31 to -1 to count backwards from the end of the month, 1 to 31 to count forward from the beginning of the month.
weekday	The day of the week that DST begins or ends: 1 for Monday, to 7 for Sunday, or 0 to ignore the weekday and use the exact date. If not zero, DST begins or ends on the specified weekday after the <i>date</i> if the date is positive, or before the <i>date</i> if negative.
hour	The hour at which DST begins or ends: 00 to 23 .

Example:

start=3/8/7/02; end=11/1/7/02

Implements the U.S. DST policy in effect since March 2007: DST begins at 2 a.m. on the second Sunday in March and ends at 2 a.m. on the first Sunday in November.

Clearing the CPE List

When the Telephony Modem loses link on all LAN interfaces, the modem clears its CPE list. This allows subscribers allowed only one CPE device to swap computers without resetting the Telephony Modem or calling support.

Provisioning Considerations

Typically, you provision the network using a PacketCable-compliant provisioning server. The server provides both provisioning tools to create data files, and servers (DHCP, DNS, TFTP) to store and transfer firmware loads and provisioning data to both the CMTS and all attached cable modems and eDVAs.

For upgrade considerations, see [Upgrading Touchstone Firmware](#) (page 26).

Interface Index Scheme

Touchstone firmware uses **ifIndex** designations to provide proper administration of multiple Gateway interfaces. The following tables list the specified defaults.

CM Interface Table

The following is the default CM interface table in AR01.1. Note that not all Touchstone devices support the maximum number of downstreams, upstreams, or SSIDs. To see the capabilities of a particular device, walk the [ifTable](#).

ifIndex	ifType	Description
1	other (1)	eRouter embedded interface
2	docsCableMacLayer (127)	RF MAC interface
3	docsCableDownstream (128)	RF Downstream interface 1
4	docsCableUpstream (129)	RF Upstream interface 1
5	usb (160)	USB interface (if supported)
6–9	ethernetCsmacd (6)	Ethernet ports 1–4
12	Removed (see note)	
16	other (1)	eDVA interface
48–78	docsCableDownstream (128)	RF Downstream interfaces 2–31
80–86	docsCableUpstream (129)	RF Upstream interfaces 2–8
200+	ipforward (142)	Logical LAN IP interfaces (see below)
300+	ipforward (142)	Logical WAN IP interfaces (see below)
10000	ieee80211 (71)	2.4 GHz WiFi radio interface
10001–10016	ieee80211 (71)	WiFi SSIDs 1–16 on 2.4 GHz radio
10100	ieee80211 (71)	5.0 GHz WiFi radio interface
10101–10116	ieee80211 (71)	WiFi SSIDs 1–16 on 5.0 GHz radio

Interfaces 200 through 207 define logical LAN subnets, as defined by DHCP. Subnets can tie together one or more physical interfaces; for example, the Ethernet ports and the subscriber's SSIDs. The [ifAdminStatus](#) for these interfaces is read-only.

Interfaces 300 through 302 define logical WAN subnets. Currently, [ifIndex](#) 300 is the CM interface as seen from the WAN, and 301 is the eRouter interface as seen from the WAN.

Interface Types

The IANAifType textual convention defines the interface types associated with entries in the [ifTable](#). Interface types used in Touchstone products include:

ifType	Description
1	Other (DOCSIS and eRouter interfaces)
6	Ethernet

ifType	Description
16	eDVA PacketCable interface
71	IEEE802.11 (wifi) interface
127	DOCSIS cable
MAC layer	
128	DOCSIS cable downstream
129	DOCSIS cable upstream
142	IP forwarding (used for logical subnets)
160	USB
198	eDVA telephony line (Voice over Cable)

Configuration File Provisioning Notes

CM and eDVA provisioning files, as described in DOCSIS and PacketCable specifications, use TLV (Type/Length/Value) objects to specify configuration parameters. This section provides information useful in provisioning Touchstone products through configuration files.

Support for TLV-41 (Downstream Channel Lists)

TLV-41 provides Downstream Channel List support. When provisioned in the CM configuration file, downstream channel lists provide the ability to specify an allowed range of downstream frequencies to use during downstream scanning operations.

When Downstream Channel Lists are specified in the configuration file, the CM does not use any frequencies outside of the provisioned range without specifically being directed to do so by the CMTS. Also, this list overrides the last operational channel value stored in NVRAM. If the CM (portion of the E-UE) loses sync with the CMTS, the CM retains the provisioned list of downstream channels provided in the configuration file, and uses them to search for a new downstream during subsequent MAC re-initialization and downstream scanning.

Full details on the operation of this feature and TLV-41 parameters can be found in Appendix C of the DOCSIS 2.0 RFI specification (SP-RFiv2.0-I11-060206).

Persistence

Some settings are optionally persistent; that is, stored in non-volatile memory in the E-UE. To clear persistent settings, reset the E-UE to factory defaults (see ["Reset to Factory Defaults"](#) (page 263) for details).

Persistence has two purposes, with different behaviors:

Changing default behavior:

Special settings may be required to address subscriber-specific issues. A common case is to disable pulse dialing when wiring or CPE issues dial "phantom" digits. Instead of creating a special configuration file, set the [arrisMtaDevEndPntDialingMethod](#) object in an SNMP browser. This type of persistent setting overrides configuration file settings.

Some objects using this type of persistence provide an **ignore** value that clears the non-volatile memory setting. This can be used to restore a factory default value without resetting the entire E-UE to factory defaults.

Setting pre-provisioning behavior:

Some settings need to configure low-level hardware before the CM component downloads its configuration file. Changes are always persistent, whether made in the configuration file or an SNMP browser, and often require a reset for the change to take effect. Once the value is stored and the reset occurs, setting the same value (in the configuration file) does not affect further operation. Changing the setting in the configuration file would again cause a hardware reset next time the CM downloads its configuration file.

Cable Modem Interface Mask

AR01.1 supports the Cable Modem Interface Mask (CMIM). The CMIM is a field in the upstream classifiers that can be used to filter out traffic based on the CM interface receiving the packet. When this field is present in the US classifier, a packet matches the classifier only if the traditional fields match and the source interface of the packet is present in the CMIM of the classifier.

The CMIM is an encoded 2- or 4-byte mask where each bit represents the interface whose [ifIndex](#) matches the bit position. The short form of the mask omits bits 16 through 31. For example, the RF interface has an [ifIndex](#) of **2** which corresponds to bit 2 of the CMIM. For this mask, bit position 0 is the most significant bit of the most significant word. For example, a CMIM classifier intended to match all of the CPE ports (external interfaces) of a CM has a CMIM value setting bits 1 and 5-15, so an encoding of either **0x47FF** or **0x47FF0000** is valid. See [Interface Index Scheme](#) (page 20) for a list of valid [ifIndex](#) values.

CM DHCP Interactions

When a Touchstone E-UE registers, the CM and eDVA make separate DHCP and TFTP requests.

The following are CM-side interactions with the DHCP server.

DHCP Option 51 Support

DHCP option 51, described in RFC 2132, allows a client device to request a particular lease time for its IP address. The option contains a 32-bit number specifying the requested lease time in seconds.

Touchstone firmware sends DHCP option 51 in DHCP Request messages during IP address renew and rebind operations.

DHCP Option 60 Support

AR01.1 uses DHCP option 60 (Vendor Class Identifier) in DHCP Discover messages to specify the DOCSIS support required. The option contains the string "docsis 3.1" to indicate DOCSIS 3.1 support.

Dual-Mode Operation

Some Touchstone devices are available in a dual-mode version that can configure its CM component for either DOCSIS or Euro-DOCSIS operation based on the type of downstream first detected.

When a dual-mode device ranges and registers for the first time, it stores the detected signal type in non-volatile memory. During subsequent reboots, it automatically scans for the stored signal type.

If a dual-mode device is moved to a plant with a different signal type, it uses several methods to detect and lock to a new signal. For example, if a dual-mode device had originally ranged and registered on a North American DOCSIS plant, and then was moved to a Euro-DOCSIS plant, it would use the following procedure:

1. Scan all cached frequencies, attempting to detect (in order):
 - QAM256 carrier using Annex A
 - QAM64 carrier using Annex A
 - QAM256 carrier using Annex B
 - QAM64 carrier using Annex B
2. Scan all preset frequencies, checking for carriers as above.
3. Perform up to three general scans, checking for carriers as above.
4. Reboot.

Forcing Provisioning Mode and Certificates

To force North American PacketCable provisioning on a dual-mode device that would otherwise default to Euro-PacketCable provisioning, set the **arrisCmDoc30SetupPacketCableRegion** object to **northAmerican(0)**. When overridden, the device continues to use the European root certificate.

Configuration Files and Signed Loads

Use European CVCs in configuration files for Dual Mode units. This allows configuration files for a normal European unit to work for Dual Mode devices as well. If the firmware load for a Dual Mode unit is signed, it should be European signed.

Certificates

Dual Mode Telephony Modems are programmed with four certificates, a European and North American certificate each for the CM and eDVA.

When the Telephony Modem boots up, it checks the value of the Dual Mode Discovered Market (DMDM) stored in NVM, and uses the certificates that correspond to that region. If the DMDM value is uninitialized, which would be the case the first time the Telephony Modem is installed or after a factory reset, it uses European signed certificates.

DMDM Status

Use the [arrisCmDevDualModeDiscoveredMarket](#) MIB object to retrieve the current DMDM value. Non-Dual Mode Telephony Modems always return **0** for this object.

Overriding the MDD IPv4/IPv6 Selection

Use the [docslf3CmMdCfgIpProvMode](#) object to override IPv4 or IPv6 selection in the MDD message. The allowed values are:

- **ipv4only(0)**: override the MDD setting and use IPv4.
- **ipv6only(1)**: override the MDD setting and use IPv6.
- **honorMdd(4)**: (default) use the IP mode set in the MDD message.

When setting this object, always use index .2 (the CATV MAC interface). Other index settings may prevent the eDVA from registering.



CAUTION

Potentially service-affecting

Override settings can potentially prevent the eDVA from registering. For example, setting this object to **ipv6(2)** when no DHCPv6 server is available causes the eDVA to attempt to register as IPv6 only, ignoring any DHCPv4 servers. Since this setting is stored in non-volatile memory, you must either change the value through SNMP or reset the Telephony Modem to factory defaults to clear the setting.



Note: In AR01.1 and newer loads, the ARRIS-proprietary object [arrisCmDoc30SetupMddIpModeOverride](#) provides the same functionality as the DOCSIS 3.0 object. Setting this object to **apm(3)** or **dpm(4)** is now equivalent to the default disable (or

honorMdd) functionality. ARRIS recommends using only one of the two objects, as setting both objects in the configuration file with conflicting values may have unexpected results.

If you set this object with an SNMP browser, and the new setting differs from the current operation mode, the CM resets to apply the new setting.

This object setting persists across reboots. Restoring the Telephony Modem to factory defaults resets the value to **disable(0)**.

Displaying the MDD Setting

You can view the current MDD setting and override using the “DHCP Parameters” troubleshooting page (select **DHCP** in the Advanced pages). This page displays both the MDD Override setting and the selected MDD mode.

The CM generates log messages in response to MDD settings or overrides:

- “MDD IP mode Set” — override disabled. The log message shows the mode set by the MDD message.
- “MDD IP mode Override” — override enabled. The log messages shows the mode set by the MIB object.

Configuring Extended Upstream Transmit Power

AR01.1 supports extended upstream transmit power capabilities, as defined in CM-SP-MULPIv3.0-I20-121113.

To enable extended upstream transmit power, set the **arrisCmDoc30SetupExtendedUpstreamTransmitPowerValue** object to the desired maximum value, in 0.25dB increments. Valid range: **205** to **244**, or **0** to disable.



Note: Not all Touchstone devices support the entire range of values. The default value of the **arrisCmDoc30SetupExtendedUpstreamTransmitPowerValue** object is the maximum supported for the device.

When enabled, Touchstone devices report extended upstream transmit power capabilities in the REG-REQ-MP message.

Upgrading Touchstone Firmware

Use this procedure to upgrade from previous versions of Touchstone firmware.

Action

Perform the following tasks as needed.

- [Upgrading the Firmware through Provisioning..... 27](#)
- [Upgrading the Firmware through SNMP..... 27](#)

Upgrading the Firmware through Provisioning

Follow these steps to upgrade the Touchstone firmware load using a provisioning server.

1. Install the new firmware on the TFTP server.
2. Use the provisioning server to add or verify the following items in the cable modem configuration file:
 - **ManufacturerCVC** (the CVC, needed only for secure downloading)
Note: only one CVC entry is allowed in a configuration file. Remove other CVC entries if they exist.
 - **UpgradeFileName** (file name of the firmware load)
 - **UpgradeServer** (IP address of the server containing the load)
 - **SnmpMib = docsDevSwAdminStatus.0 2** (allowProvisioningUpgrade)
3. During the maintenance window, use your provisioning server or element manager to reset each Touchstone E-UE.
The E-UEs download the new firmware, then reset.
4. Verify that the E-UE has the new load by checking the value of the **docsDevSwOperStatus** object (using an SNMP server).
The value should read **completeFromProvisioning(3)**.

Upgrading the Firmware through SNMP

Follow these steps to upgrade the Touchstone firmware load using an SNMP manager.

1. Using the provisioning server, add the ManufacturerCVC to the configuration file.
Note: only one CVC entry is allowed in a configuration file. Remove other CVC entries if they exist.
2. Using the SNMP manager, set the following **docsDevSoftware** objects:
 - docsDevSwServerAddressType**
Set to **1** for IPv4 server addressing or **2** for IPv6 addressing.
 - docsDevSwServerAddress**
The IP address of the server containing the load.
 - docsDevSwFilename**
The file name of the load.
 - docsDevSwAdminStatus**
Set to **upgradeFromMgt(1)**.
The E-UE downloads the new firmware, then resets.
3. Verify that the E-UE has the new load by checking the value of the **docsDevSwOperStatus** object.
The value of the object should read **completeFromMgt(3)**.

Configuring Channel Bonding Characteristics

Channel bonding is set up in the CM configuration file. The MIB objects in this procedure provide some extra control over, and monitoring of, channel bonding.

Action

Follow these steps to configure channel bonding characteristics.

1. To disable downstream channel bonding (effectively forcing DOCSIS 2.0 behavior), set the **arrisCmDoc30SetupDsBonding** object to **disable(0)**.



Note: The object setting takes effect at the next reboot, and persists across reboots. If this object is set to **disable(0)**, the modem disables DOCSIS 3.0 operation until re-enabled.

2. To configure how the Telephony Modem handles partial service situations, proceed to [Recovery from Partial Service](#) (page 207).
3. To display the current bonding mode, query the **arrisCmDoc30BondingMode** object. This object contains a string showing the current DOCSIS operating mode and the number of bonded downstream and upstream channels.

Examples:

```
DOCSIS3.0 4x1
DOCSIS2.0 1x1
```

IPv6 Provisioning Notes

This section describes provisioning modes and flows appropriate to IPv6 support.

IPv6 Provisioning Modes

Touchstone firmware supports the following provisioning modes for IPv6:

- SECURE (full PacketCable)
- BASIC.1/2 (PacketCable and ARRIS versions)
- HYBRID.1/2
- PacketCable Minus KDC

Single MAC provisioning is explicitly not supported.

Selecting an Addressing Mode

Touchstone firmware supports the DOCSIS 3.0 MAC Domain Descriptor (MDD) message, defined in the *MAC and Upper Layer Protocols Interface Specification*, CM-SP-MULPIv3.0-I20-

121113, for selecting IPv4 or IPv6 addressing. The CM uses TLV 5.1 in the MDD to select the addressing mode as follows:

Value	Addressing Mode
0	IPv4
1	IPv6
2	Alternate Provisioning Mode (APM): try IPv6 first, then IPv4
3	Dual Provisioning Mode (DPM)

If the E-UE does not find an MDD during provisioning, it always selects IPv4 addressing.

The [docsIf3CmMdCfIpProvMode](#) object can override the MDD and set IPv4 or IPv6 operation.



CAUTION

Potentially service-affecting

Use this feature carefully. An invalid setting could isolate the Telephony Modem from the network.

The supported values are:

- **ipv4only(0)**: force IPv4 addressing
- **ip6only(1)**: force IPv6 addressing
- **honorMdd(4)**: Use the MDD to determine the address type

The [arrisCmDoc30SetupMddIpModeOverride](#) MIB object can override the MDD and set IPv4 or IPv6 operation.

DHCP Behavior for IPv6 Provisioning

When the CM receives an MDD message that specifies IPv6 operation, it acquires its IP address according to the *MAC and Upper Layer Protocols Interface Specification*, CM-SP-MULPIv3.0-I20-121113. AR01.1 supports both SLAAC and DAD mechanisms.

AR01.1 supports the DHCPv6 options listed in the following table. For details, see the *DOCSIS 2.0 + IPv6 Cable Modem Technical Report*, CM-TR-DOCSIS2.0-IPv6-V01-080307.

Option #	Sub-Option	Name
1		Client Identifier option (DUID)
2		Server Identifier Option
3		IA_NA option (IPv6 address)
6		Option Request Option
14		Rapid Commit Option

Option #	Sub-Option	Name
19		Reconfigure Message option
20		Reconfigure Accept Option
17		Vendor-specific information option
	32	TFTP Server Addresses option
	33	Configuration File Name option
	34	Syslog Server Addresses option
	35	TLV5 Encoding
	36	DOCSIS Device Identifier option
	37	Time Protocol Servers option
	38	Time Offset option

AR01.1 supports the DHCP Reconfigure message described in RFC 3315. Upon receiving a DHCP Reconfigure message, the CM validates the message then acquires updated DHCP parameters from the server.

Provisioning File Notes

The configuration file must be specific to either IPv6 or IPv4. If the plant has mixed IPv4 and IPv6 CMs, each address type requires separate provisioning files.

For IPv6 configuration, addresses must be fully qualified and not compressed. For example, an IPv6 address of **2001:0200:0000:0000:0000:0000:0022** cannot be entered as **2001:0200::0022**.

TLV-38 Enhancements

AR01.1 supports TLV-38 (Notification) sub-type 8 (SNMP notification IPv6 address). Specify this sub-TLV to send SNMP traps and informs to an IPv6-configured receiver.

Configuring the Diplexer

Some Touchstone DOCSIS 3.1 devices allow diplexer configuration, to adjust the upstream and downstream frequency ranges to accommodate the HFC plant.

The diplexer capabilities depend on the market as follows:

North American models:

- Upstream: 5 MHz to 42 MHz, or 5MHz to 85 MHz
- Downstream: 108 MHz to 1002 MHz (fixed)

European models:

- Upstream: 5 to 85 MHz, or 5 to 204 MHz
- Downstream: 108 to 1218 MHz, or 258 to 1218 MHz

To configure the diplexer:

1. Read the **arrisCmDoc30DiplexerFrequencyRanges** object to determine which frequency ranges the device supports. The result is a string, showing the upstream/downstream ranges for each band, similar to the following:
Band0: 5- 85MHz/108- 1002MHz; Band1: 5- 42MHz/108- 1002MHz
2. Set the **arrisCmDoc30DiplexerControl** object to the desired band: **0** for Band 0, **1** (the default) for Band 1.
3. Reset the Touchstone device for the change to take effect.

The setting persists across reboots, and is not affected by a reset to factory defaults.

NCS Voice Provisioning

All Touchstone Telephony Modems and Telephony Gateways provide telephony service through an eMTA (also known as eDVA) component. This chapter provides information for provisioning NCS telephony.

NCS Provisioning Considerations

About IPsec

IPsec (Internet Protocol Security) is a collection of Internet standards used to encrypt and authenticate IP packets, to provide message integrity and privacy. IPsec provides security at the network layer (all TCP and UDP packets, and layers above).

IPsec is controlled by setting the `pkcMtaDevCmsIpsecCtrl` object for each CMS that the eDVA can communicate with; you can include this object in the eDVA configuration file. The object is indexed by the CMS FQDN for North American loads. Set the object to `true(1)` to enable IPsec between the eDVA and a particular CMS, and `false(2)` to disable it.



Note: Touchstone E-UEs use only the IPsec ESP transport mode.

Call Management Servers

Touchstone firmware accepts up to 64 call management server IP addresses identified in the eDVA configuration file. Each call server DNS entry can have up to six IP addresses associated with it, so assigning multiple IP addresses to a CMS reduces the total number of unique servers that can be listed. Support for multiple CMSs allows for load balancing, where an eDVA can be redirected to use a CMS with a lighter load.

When IPsec is activated, Touchstone E-UEs store up to 10 security associations, limiting the number of CMSs that it can communicate with at any given time. However, by setting the “CMS Redirect” bit (0x00400000) in the CallP Feature Switch (see [CallP Feature Switch](#) (page 40)), and by listing up to 9 CMSs in the configuration file, the Telephony Modem can bypass the 10-CMS limit, and support redirection to any other CMS on the customer network.

Voice and Signaling Ports

AR01.1 firmware uses a random selection of ports in the range 49152 through 65535 for RTP- and RTCP-based voice communications. The port numbers can be modified using the `arrisMtaCfgRTPDynPortStart` and `arrisMtaCfgRTPDynPortEnd` objects.

By default, the eDVA uses port 2727 on the upstream, and port 2427 on the downstream, to send and receive signaling information. You can change the default port number in the eDVA configuration file. You can also change the transmit port by sending an NCS message from the call server once the eDVA is operating.

eDVA Interface Table

The default **ifTable** for the eDVA is:

ifIndex	ifType	Description
1	other (1)	DOCSIS Embedded Interface
9	voiceOverCable (198)	Telephony Line 1
10	voiceOverCable (198)	Telephony Line 2
11	voiceOverCable (198)	Telephony Line 3 (if available)
12	voiceOverCable (198)	Telephony Line 4 (if available)

Provisioning Modes

Touchstone firmware supports PacketCable provisioning modes.

PacketCable Provisioning Modes

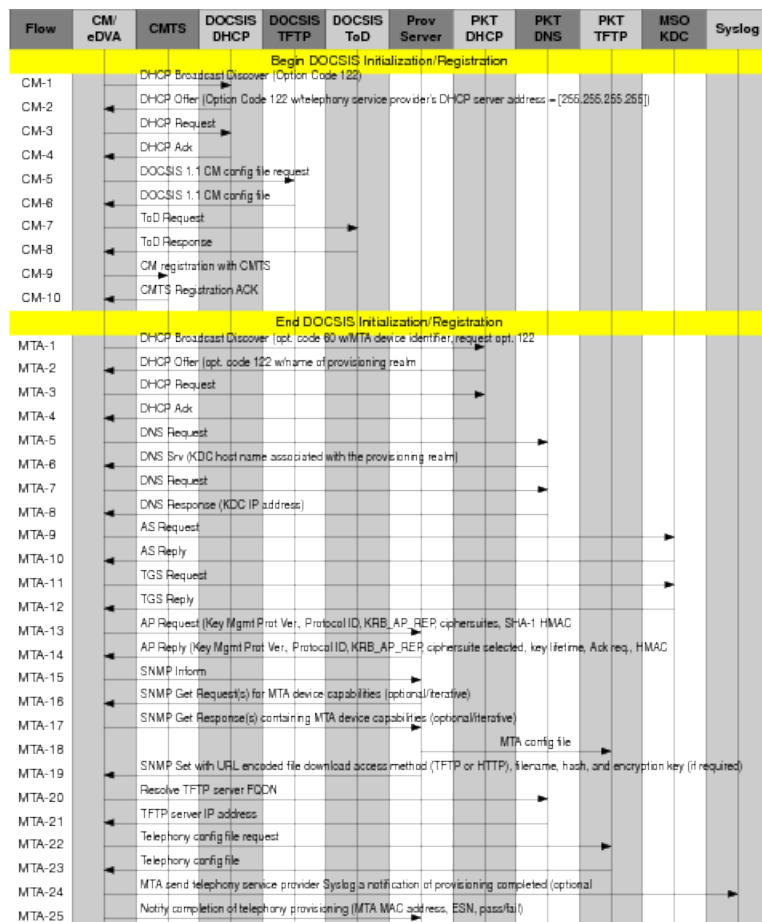
AR01.1 supports the standard PacketCable provisioning modes.

PacketCable SECURE (Full PacketCable)

This mode is also called “Full PacketCable,” and is the default provisioning mode. The data and telephony components have unique IP addresses, MAC addresses, and configuration files (i.e. two of each per E-UE). When the E-UE registers, it makes two separate DHCP and TFTP requests.

SNMP communication uses SNMPv3, sending an SNMPv3 INFORM. The E-UE and provisioning system support Kerberos mutual authentication and Kerberized SNMPv3 messaging.

IPsec is supported, and may be enabled or disabled using the `pkcMtaDevCmsIpsecCtrl` object (enabled by default). Media encryption (voice security) can be enabled on a per-call basis using NCS signaling (the LCO/SDP options) or disabled per eDVA using a feature switch. The feature switch is stored in NVRAM. The following diagram shows the full PacketCable event sequence. All events are included.

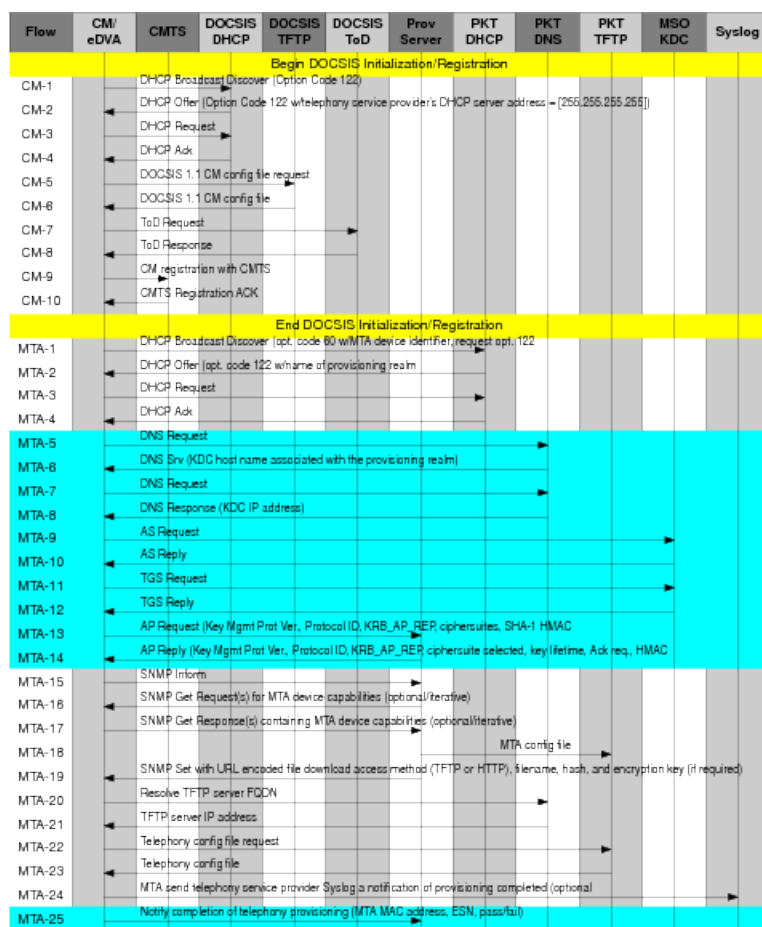


PacketCable HYBRID

Similar to the ARRIS-proprietary “PacketCable without KDC” provisioning mode. HYBRID flows are identical to the SECURE flow but remove the Kerberos message exchange, and use SNMPv2c instead of SNMPv3.

There are two HYBRID flows, HYBRID.1 and HYBRID.2; the primary difference between the two is that HYBRID.2 uses the “provisioning complete” SNMP INFORM. IPsec is disabled.

Media encryption can be controlled on a per-eDVA basis using a feature switch. The following diagram shows the PacketCable HYBRID event sequence. This sequence skips several events in the eDVA provisioning; the shaded steps below are skipped.



Note: Only HYBRID.1 skips step MTA-25.

PacketCable BASIC

Simplified provisioning flows with no Kerberos or SNMPv3 security, and no SNMP enrollment using SNMP INFORM.

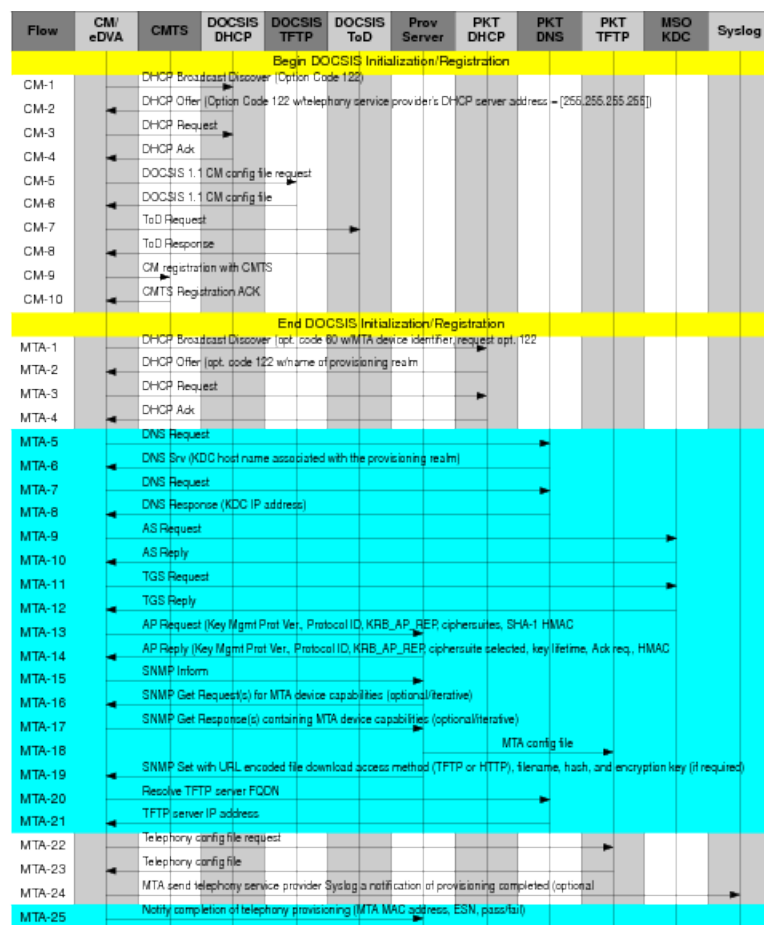
There are two BASIC flows, BASIC.1 and BASIC.2; the primary difference between the two is that BASIC.2 uses the “provisioning complete” SNMP INFORM.

When using a PacketCable BASIC mode, the downloaded configuration file must contain the MIB object **pktcMtaDevProvConfigHash**. The eDVA calculates the hash value of the provisioning file and verifies that the calculated hash and the hash value contained in the MIB object match. If they do not match, provisioning fails.



Note: PacketCable BASIC requires that the eDVA provisioning file contain the **pktcMtaDevProvConfigHash** object, with a value equal to the hash of the provisioning file. The eDVA calculates the hash and compares it to the value of this object; if the values do not

match, provisioning fails. The following diagram shows the sequence for PacketCable BASIC.1. This sequence skips several steps in the eDVA provisioning. The PacketCable BASIC.2 sequence is nearly identical to BASIC.1, but does not skip the last step in the eDVA provisioning.



Verifying eDVA Provisioning and Endpoint Status

The **pktcMtaDevProvisioningState** object indicates the status of the eDVA initialization process. The MIB object **pktcNcsEndPntStatusError** indicates whether the endpoint has successfully registered with the call server.

DHCP Support by Provisioning Mode

The following sections list DHCP parameters used by each provisioning mode.

Options Required in All Provisioning Modes

The following DHCP options are required in all CM and eDVA offers. The E-UE cannot function without a subnet mask and at least one router, DNS server, and Syslog server.

Option	Description
1	Subnet mask
3	IP address of the gateway router (one or more)
6	IP address of the DNS servers (one or more)
7	IP address of the log servers (one or more)



Note: In addition to the required options listed above, ARRIS strongly recommends including option 4 (IP address of the ToD server) in all CM and eDVA offers.

PacketCable Modes

These options are valid for Full PacketCable and PacketCable minus KDC provisioning modes. Note that options 122 and 177 are mutually exclusive (specify one or the other, not both). Option 177 is the default for PacketCable minus KDC.

CM DHCP Option 4: ToD Server IP Address

CM DHCP Option 122:

- SubOption 1: Service Provider's Primary DHCP (required)
- SubOption 2: Service Provider's Secondary DHCP (optional)

eDVA DHCP Option 122:

- SubOption 3: Service Provider's SNMP Entity (required)
- SubOption 4: AS REQ/REP Exchange Backoff and Retry for SNMPv3 Key Management (optional)
- SubOption 5: AP REQ/REP Exchange Backoff and Retry for SNMPv3 Key Management (optional)
- SubOption 6: Kerberos Realm (FQDN) (Full PacketCable, Basic.1, Basic.2: required)
- SubOption 7: Authorization method (TGT for eDVA) (optional)
- SubOption 8: Provisioning Timer (minutes) (optional)
- SubOption 9: Security Ticket Invalidation (optional)

CM DHCP Option 177:

- SubOption 1: Service Provider's Primary DHCP (required)
- SubOption 2: Service Provider's Secondary DHCP (optional)

eDVA DHCP Option 177:

- SubOption 3: Service Provider's SNMP Entity (required)
- SubOption 4: Service Provider Network Primary DNS
- SubOption 5: Service Provider Network Secondary DNS
- SubOption 6: Kerberos Realm (FQDN)
- SubOption 7: Authorization method (TGT for eDVA)
- SubOption 8: Provisioning Timer (minutes)



Note: By default, the E-UE performs a “reinit” when it receives an SNMP Entity (sub-option 3) that differs from the original entity IP address. The [arrisMtaDevDhcpOptionOverride](#) MIB object allows you to provision the E-UE to accept a changed SNMP entity. This may be necessary for operation with certain DHCP servers that reassign SNMP entities for load-balancing.

eDVA DHCP Interactions

When a Touchstone E-UE registers, the CM and eDVA make separate DHCP and TFTP requests.

The Telephony Modem eDVA component sends various information to the provisioning server using the DHCP options described below.

DHCP Option 43 Support

AR01.1 sends DHCP option 43 (Vendor-Specific Information) in eDVA DHCP Discover messages with the following sub-options:

Sub-opt.	Name	Value
2	Device Type	“EDVA”
4	Serial Number	Varies (e.g. “997BNW87D747320”)
5	HW Version	Hardware version of eDVA (e.g. “2.0”)
6	SW Version	Firmware version (e.g. “9.1.115”)
7	Boot ROM	Boot ROM version (e.g. “1.2.1.20”)
8	Vendor ID	“0000CA”
9	Model Number	E-UE model number (e.g. “TM2472A”)
10	Vendor Name	“Arris Interactive, L.L.C.”
31	MTA MAC Addr.	Varies (e.g. “0015d004153d”)
32	Correlation ID	Varies (must match the value of the pktcMtaDevCorrelationId object).

DHCP Option 51 Support

Touchstone firmware sends DHCP option 51 in DHCP Request messages during IP address renew and rebind operations.

Disabling Option 122 Sub-Option 3 Enforcement

Touchstone firmware can be configured to ignore the “SNMP Entity” (DHCP Option 122 Sub-Option 3) comparison checks during eDVA DHCP Renew and Rebind processing. This may be necessary if you want to allow the eDVA to direct Link Up and Link Down traps to an alternate trap server.

To configure this setting, set the `arrisMtaDevDhcpSubOpt3Immediate` object to `off(1)` to disable the override, or `on(2)` to enable the override.

Provisioning Quality of Service

DOCSIS and PacketCable standards specify the use of service flows to separate and prioritize voice and data traffic. Touchstone firmware provides two QoS modes.

See CM-SP-MULPIV3.1 for detailed descriptions of Service Flow types.

Full DQoS Mode

Touchstone firmware defaults to Dynamic Quality of Service (DQoS) provisioning. Full DQoS simplifies provisioning by requiring only that the primary Best Effort (BE) and MGCP (signaling) flows be provisioned. The firmware dynamically sets up and tears down UGS service flows, using a standard set of parameters designed for efficient use in DOCSIS-based networks, as needed. The CMS controls the bandwidth authorization as specified in the PacketCable *Dynamic Quality of Service* Specification, PKT-SP-DQOS1.5-I03-070412.

Full DQoS provides an added layer of security by authenticating eDVAs that contact the CMS during call setup. Each session is authorized; the session authorization uses a handle (the Gate-ID) assigned by the CMTS, passed to the CMS, and sent to the eDVA using an NCS message, to match requests with authorizations. Upon receiving call-signaling information, the eDVA passes the Gate-ID to the CMTS in a DSA/DSC message.

DSX QoS Mode

Touchstone firmware supports an ARRIS-proprietary feature that implements QoS using UGS flows for voice transmission using DOCSIS 1.1 DSx messaging. This functionality provides a level of QoS in a network where the CMS and CMTS do not support the PacketCable Full DQoS model.

DSx QoS functionality can be activated using a feature switch. When activated, the firmware sends the appropriate DSx messages needed to Add/Modify/Delete the UGS service flows. DSx messages flow only between the CMTS and the eDVA, and do not involve the CMS in any validation or requests for setting up or monitoring the UGS flows.



Note: When using this functionality with the ARRIS C4 CMTS, PacketCable authorization needs to be disabled. Contact your next level of support for instructions.

Feature Switches

Touchstone firmware provides various feature switches to enable or disable various functionality. Some feature switches provide improved interoperability with non-PacketCable compliant equipment, while others provide features that may violate DOCSIS or PacketCable standards.

CallP Feature Switch

Touchstone firmware provides an ARRIS-specific object, [ppCfgMtaCallPFeatureSwitch](#), used to configure the Telephony Modem for the specific sub-set of PacketCable features supported by the selected network configuration. This allows the flexibility to interoperate with other vendors by providing the ability to enable or disable the proper functionality. The default is full PacketCable compatibility.

The feature switch is a 32-bit value, where each bit enables or disables a certain feature. Most of these values should only be changed with the guidance of ARRIS technical support, but some flags may be changed as necessary.

The following is a list of CallP Feature Switches that can be adjusted at your discretion. The default value is **0x0**.



Note: SIP loads support a subset of the available bit values. The following table indicates whether each bit value is available only for NCS or for both NCS and SIP loads.

Bit	Description
0x00000001	<p>Disable NCS Piggyback Messages (NCS only)</p> <p>Set this bit to disable transmission of NCS piggybacked messages (that is, sending more than one NCS message in a UDP packet). This may be required by Call Agents that do not properly handle piggybacked NCS messages.</p> <p>Note: NCS redirection may not function properly when piggybacked messages are disabled.</p>
0x00000002	<p>Prevent Endpoint Lockstep Quarantine Mode (NCS only)</p> <p>Set this bit to prevent endpoints from entering the lockstep quarantine mode. This may be required by Call Agents that leave the endpoint in lockstep mode.</p> <p>When this bit is clear, the gateway must receive a new Notification Request command after sending a Notify command. Until this happens, the endpoint is in a lockstep state, and events that occur and are to be detected are simply stored in the quarantine buffer until receiving the Notification Request command.</p>
0x00000004	<p>Show OOS instead of Idle for unprovisioned lines</p> <p>Set this bit to return oos(0) instead of idle(1) in the arrisMtaDevLineCardState object for unprovisioned lines.</p>
0x00000008	<p>T.38 Capability Descriptor (NCS only)</p> <p>Set this bit to reduce the SDP capability descriptor to only send T.38-related information.</p>

Bit	Description
0x00000040	Automatic OSI (NCS only) Set this bit to automatically apply OSI to both the originating and terminating sides of a call. To fully enable this functionality, set bit 0x20000000 as well.
0x00000080	Omit MPTIME parameter in returned SDP (NCS only) Set this bit to omit the “a=mptime” and “a=ptime” parameters in the eDVA’s SDP message.
0x00000100	Enable DOCSIS 2.0 backward compatibility for DQoS (NCS and SIP) Set this bit to enable DOCSIS 3.0 or newer Telephony Modems to properly interoperate with a DOCSIS 2.0 CMTS.
0x00000800	Force use of DQoS (NCS only) Set this bit to force the Telephony Modem to require Dynamic Quality of Service, rejecting service otherwise. The default behavior allows non-DQoS service.
0x00001000	Nuera RFC 2833 messaging without request using payload 127 (NCS only) Set this bit to instruct the eDVA to generate RFC 2833 events with a payload type of 127 without specifically being instructed to do so (for compatibility with the Nuera RDT). Clear this bit (the default) to generate RFC 2833 events using NCS signaling and SDP exchange.
0x00004000	DSx/Access only DQoS (NCS and SIP) Set this bit to use DSx/Access-only DQoS only between the CM and CMTS.
0x00008000	Disable endpoint from sending provisional responses (NCS only) Set this bit to disable sending provisional responses to the CMS if execution of the CRCX or MDCX commands takes additional time to execute. The ARRIS eDVA sends this provisional response if DQoS is to be performed, due to the extra amount of time it takes to set up bandwidth between the CM and CMTS. Once the provisional response is issued, the CMS should stop retransmitting the command. When the eDVA has completed the transaction, it transmits a “final” response back to the CMS. This “final” response must be acknowledged by the CMS; otherwise, the eDVA retransmits it until it is acknowledged. The default is to send provisional responses in accordance with PacketCable ECN MGCP-N-02218. Set this bit to provide compatibility with CMS vendors that are not capable of supporting provisional responses.
0x00010000	Payload Header Suppression (NCS and SIP) Set this bit to allow Payload Header Suppression of voice packets between the CM and CMTS. Note: This bit only affects PHS for RTP voice packet streams. Its setting does not affect PHS for DOCSIS data packets, which is controlled through the DOCSIS MIB. Conversely, DOCSIS MIB settings do not affect PHS for RTP voice packets.

Bit	Description
0x00080000	<p>LUCENT RFC-2833 messaging without request using payload 94 (NCS only)</p> <p>Some CMS vendors use RFC 2833 to have the eDVA pass detected telephony events (e.g. offhook, onhook, digits) to a PSTN gateway in-band, similar to ABCD robbed-bit signaling. For compatibility with Lucent iMerge, set this bit to instruct the eDVA to generate RFC 2833 events with a payload type of 94 during call setup, without specifically being instructed to do so. Clear this bit (the default) to generate RFC 2833 events using NCS signaling and SDP exchange.</p>
0x00100000	<p>Allow AES encryption for RTP/RTCP (NCS only)</p> <p>Clear this bit (the default) to allow the negotiation of voice encryption, which is a requirement of PacketCable, and is enabled by default on a per call basis. Set this bit to disable this feature, and reduce the size of the eDVA's SDP message since encryption parameters are not included.</p> <p>Note: If the CMS (and far end) can handle the increased size of the SDP with AES encryption enabled, then set this bit to 0, as the far end is capable of negotiating voice security parameters (including NULL encryption) and the CMS will instruct the eDVA on whether to use encryption or not on a per-call basis.</p>
0x00400000	<p>NCS Redirect without IPsec (NCS only)</p> <p>Set this bit to allow a CMS to redirect the eDVA to another CMS that is not provisioned in the eDVA CMS table, allowing the eDVA to communicate with the CMS without attempting to establish an IPsec association first. Setting this bit is for redirect cases only; the eDVA does not respond to call servers not provisioned in the eDVA's CMS table.</p>
0x00800000	<p>Add brackets around IP for eDVA FQDN (NCS only)</p> <p>Set this bit to enable the eDVA to send an NCS message with a bracketed IP address as part of the eDVA FQDN when communicating with the call server.</p> <p>For call servers that use IP address information instead of FQDNs, the brackets surrounding the IP address are mandatory.</p> <p>The following example shows the messaging format with this feature switch enabled.</p> <pre>aal n/1@[10. 10. 13. 11] MGCP 1.0 NCS 1.0</pre>
0x01000000	<p>Send DTMF digits via RFC 2833 with operator-specified payload without request (NCS only)</p> <p>Set this bit to instruct the eDVA to generate RFC 2833 DTMF events using a payload type defined in the ppCfgrfc2833DigitPayloadType MIB object without being instructed to do so by the CMS. The default payload type value is 101.</p>
0x08000000	<p>Use alternate (non-sequential) Caller ID delivery order (NCS and SIP)</p> <p>Set this bit to have the eDVA present Caller ID in an alternate (non-sequential) order. This may be required for compliance with some CPE devices that expect Caller ID information to be presented in a non-standard format. Clear this bit (the default) to use a sequential order based on the parameter type (date/time, then number, then name).</p>

Bit	Description
0x10000000	<p>Delay DLCX against connection on on-hook only line (for VMWI) (NCS only)</p> <p>Set this bit to delay processing of a Delete Connection message from the CMS for a line that is on-hook, to allow any queued RTP packets in the DSP jitter buffer to be played out. In the case of VMWI, this prevents the DSP connection from being closed while delivering FSK signals for VMWI. The default setting for this bit is cleared (0).</p>
0x20000000	<p>Enable Automatic OSI (NCS only)</p> <p>Set this bit to have the eDVA automatically generate OSI (Open Switch Interval) to the CPE upon far end termination of a call (i.e. the eDVA receives a DLCX command for the last connection on the endpoint).</p> <p>The arrisMtaDevAutomaticOsiDelay object specifies the delay, in 100 ms increments, before sending the OSI. The eDVA cancels OSI generation if any of the following events occur before the timer expires:</p> <ul style="list-style-type: none"> ■ line goes on-hook ■ hook flash on the line ■ a new connection is created on the line ■ CMS receives OSI request <p>The valid range for the MIB object is 0 (no delay) to 100 (10 seconds).</p>

For non-PacketCable configuration settings, contact your ARRIS Technical Support representative.

For more information, see the *PacketCable Network-Based Call Signaling Protocol Specification*, PKT-SP-NCS1.5-I04-120412.

Example

If your configuration requires DSx-QoS, set the feature switch to include the 0x4000 and 0x10000 flags, using PacketACE or a provisioning server. If no other flags are required, the setting would be as follows:

```

Snmplib = ppCfgMtaCallpFeatureSwitch.0 hexstr: 00.01.40.00 = 00.00.00.00
(default)
+ 00.01.40.00 (additional features)

```

CallP Feature Switches Affecting the SDP

The following feature switches affect the SDP, returned in response to a Create (CRCX) or Modify (MDCX) Connection command.

Bit	Description
0x00000008	Reduce the capability descriptor in the SDP to T38 only (default = 0, no reduction).
0x00000080	Omit mptime parameter in returned SDP (default = 0,

Bit	Description
	mptime included).
0x00001000	NUERA RFC 2833 messaging without request using payload 127 (default = 0, telephone-event is negotiated normally).
0x00080000	LUCENT RFC 2833 messaging without request using payload 94 (default = 0, telephone-event is negotiated normally).
0x00100000	Allow AES encryption for RTP/RTCP (default = 1, AES encryption is negotiated normally).
0x01000000	Send DTMF digits via RFC 2833 with an operator-defined payload without request (default = 0, telephone-event is negotiated normally).

The following CRCX message is used to generate all the SDP examples, unless otherwise specified:

```
CRCX 19901 aal n/1@mta218. dev36 MGCP 1.0 NCS 1.0
C: 1234
M: recvonly
L: mp: 20, a: PCMU, fxr/fx: t38-loose, xrm/mcr: on
```

The default feature switch settings are **(0x0)** for NCS loads, and **0x0020** for SIP loads) generate the following SDP:

```
v=0
o=- 381749076 381749076 IN IP4 10.1.36.219
s=-
c=IN IP4 10.1.36.219
t=0 0
m=audio 65496 RTP/AVP 0
b=AS: 81
a=rtcp-xr: voip-metrics
a=mptime: 20
a=rtpmap: 0 PCMU/8000/1
a=X-pc-secret: base64: ZNos/qs530eSDJ4FvdL2GJBR62lS5UKyQ7n9og4
IaadbA9Blpg6lM2Pf0aHEGg== U5Q5N/eWni mq9Q/yj WwY2hACRI Y6a9qqEQ
Us8tm54lEmEE6LXkCB51+3sqxlQg==
a=X-pc-suites-rtp: 62/51 64/51 60/51 60/50
a=X-pc-suites-rtcp: 81/70 81/71 82/70 82/71 80/70
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 8 15 18 96 97 98 2 99 101
a=cpar: a=fmtp: 18 annexb=no
a=cpar: a=rtpmap: 101 telephone-event/8000/1
a=cpar: a=fmtp: 101 0-15, 144, 149, 159
a=cdsc: 11 image udptl t38
a=cpar: a=T38FaxVersion: 0
a=cpar: a=T38FaxRateManagement: transferredTCF
a=cpar: a=T38FaxMaxDatagram: 161
a=cpar: a=T38FaxUdpEC: t38UDPRedundancy
```

When the “Reduce Capability Descriptor” switch (**0x00000008**) is enabled, and all other switches are set to their default values, the SDP becomes:

```
v=0
o=- 381749076 381749076 IN IP4 10.1.36.219
s=-
c=IN IP4 10.1.36.219
```

```

t=0 0
m=audio 65496 RTP/AVP 0
b=AS: 81
a=rtcp-xr: voip-metrics
a=mptime: 20
a=rtpmap: 0 PCMU/8000/1
a=X- pc- secret: base64: ZNos/qs530eSDJ4FvdL2GJBR62lS5UKyQ7n9og4
IaadbA9Blpg6lM2Pf0aHEGg== U5Q5N/eWni mq9Q/yj WwY2hACRI Y6a9qqEQ
Us8tm54lEmEE6LXkCB5l+3sqxlQg==
a=X- pc- csuites- rtp: 62/51 64/51 60/51 60/50
a=X- pc- csuites- rtcp: 81/70 81/71 82/70 82/71 80/70
a=sn: 0
a=cdsc: 1 image udptl t38

```

In this example, the Capability Descriptor is reduced to only the information required to relay support for UDPTL T.38 to the far end. This allows the far end to support T.38 strict mode as defined in the PacketCable 1.5 NCS specification.

When the “Omit mptime” switch (**0x00000080**) is enabled, and all other switches are set to their default values, the SDP becomes:

```

v=0
o=- 382093395 382093395 IN IP4 10.1.36.219
s=-
c=IN IP4 10.1.36.219
t=0 0
m=audio 58810 RTP/AVP 0
b=AS: 81
a=rtcp-xr: voip-metrics
a=rtpmap: 0 PCMU/8000/1
a=X- pc- secret: base64: XI51bgXR5MNudaKXi sS0tj YCc90x3f7j A+oj yam
W/O/M2BlCaejlrRL0dApR6w== 6LbN8ULCFGmj cR2T3l 1uZuBcfWM2vfGn09
YTmR60hHfQwC4eE+WWSX7AarnFPA==
a=X- pc- csuites- rtp: 62/51 64/51 60/51 60/50
a=X- pc- csuites- rtcp: 81/70 81/71 82/70 82/71 80/70
a=sn: 0
a=cdsc: 1 audio RTP/AVP 0 8 15 18 96 97 98 2 99 101
a=cpar: a=fmtp: 18 annexb=no
a=cpar: a=rtpmap: 101 telephone-event/8000/1
a=cpar: a=fmtp: 101 0-15, 144, 149, 159
a=cdsc: 11 image udptl t38
a=cpar: a=T38FaxVersion: 0
a=cpar: a=T38FaxRateManagement: transferredTCF
a=cpar: a=T38FaxMaxDatagram: 161
a=cpar: a=T38FaxUdpEC: t38UDPRedundancy

```

In this example, neither the `a=ptime` nor the `a=mptime` parameters are included in the SDP since the packetization rate is the default (20 ms). If the Call Agent had specified a different packetization rate, then the `a=ptime` parameter is included as follows:

```

v=0
o=- 382093395 382093395 IN IP4 10.1.36.219
s=-
c=IN IP4 10.1.36.219
t=0 0
m=audio 58810 RTP/AVP 0
b=AS: 81
a=rtcp-xr: voip-metrics
a=ptime: 10
a=rtpmap: 0 PCMU/8000/1
a=X- pc- secret: base64: XI51bgXR5MNudaKXi sS0tj YCc90x3f7j A+oj yam
W/O/M2BlCaejlrRL0dApR6w== 6LbN8ULCFGmj cR2T3l 1uZuBcfWM2vfGn09

```

```

YtmR60hHfQwC4eE+WWSX7AarnFPA==
a=X- pc- csuites- rtp: 62/51 64/51 60/51 60/50
a=X- pc- csuites- rtcp: 81/70 81/71 82/70 82/71 80/70
a=sn: 0
a=cdsc: 1 audio RTP/AVP 0 8 15 18 96 97 98 2 99 101
a=cpar: a=fmtp: 18 annexb=no
a=cpar: a=rtpmap: 101 telephone- event/8000/1
a=cpar: a=fmtp: 101 0- 15, 144, 149, 159
a=cdsc: 11 image udptl t38
a=cpar: a=T38FaxVersion: 0
a=cpar: a=T38FaxRateManagement: transferredTCF
a=cpar: a=T38FaxMaxDatagram: 161
a=cpar: a=T38FaxUdpEC: t38UDPRedundancy

```

When the “Allow AES Encryption” switch (**0x00100000**) is disabled, and all other switches are set to their default values, the SDP becomes;

```

v=0
o=- 381749076 381749076 IN IP4 10. 1. 36. 219
s=-
c=IN IP4 10. 1. 36. 219
t=0 0
m=audio 65496 RTP/AVP 0
b=AS: 81
a=rtcp- xr: voi p- metri cs
a=mptime: 20
a=rtpmap: 0 PCMU/8000/1
a=sn: 0
a=cdsc: 1 audio RTP/AVP 0 8 15 18 96 97 98 2 99 101
a=cpar: a=fmtp: 18 annexb=no
a=cpar: a=rtpmap: 101 telephone- event/8000/1
a=cpar: a=fmtp: 101 0- 15, 144, 149, 159
a=cdsc: 11 image udptl t38
a=cpar: a=T38FaxVersion: 0
a=cpar: a=T38FaxRateManagement: transferredTCF
a=cpar: a=T38FaxMaxDatagram: 161
a=cpar: a=T38FaxUdpEC: t38UDPRedundancy

```

The last three CPFS bits (**0x00001000**, **0x00080000**, and **0x01000000**) are related to RFC 2833. They are used in specific configurations and are designed to skip CODEC negotiation. For example, when the “Nuera RFC2833” feature is enabled, and all other switches are set to their default values, the SDP becomes:

```

v=0
o=- 382250430 382250430 IN IP4 10. 1. 36. 219
s=-
c=IN IP4 10. 1. 36. 219
t=0 0
m=audio 63672 RTP/AVP 0
b=AS: 81
a=rtcp- xr: voi p- metri cs
a=mptime: 20
a=rtpmap: 0 PCMU/8000/1
a=X- pc- secret: base64: Qb8GFLNXP4Z3yi yxFx1Ws9vLph9qG6bTIXezUl z
rwi a7i iNvPPkVYdZhZ77NEQ== zZj gwXRR2j 5F04l DXefPTV06PT8g31Hn5V
Ea6NJvFPFsPi raDeDI 35EI 8K0+4A==
a=X- pc- csuites- rtp: 62/51 64/51 60/51 60/50
a=X- pc- csuites- rtcp: 81/70 81/71 82/70 82/71 80/70
a=sn: 0
a=cdsc: 1 audio RTP/AVP 0 8 15 18 96 97 98 2 99 127
a=cpar: a=fmtp: 18 annexb=no
a=cpar: a=rtpmap: 127 telephone- event/8000/1
a=cpar: a=fmtp: 127 0- 15, 144, 149, 159
a=cdsc: 11 image udptl t38

```

```

a=cpar: a=T38FaxVersion: 0
a=cpar: a=T38FaxRateManagement: transferredTCF
a=cpar: a=T38FaxMaxDatagram: 161
a=cpar: a=T38FaxUdpEC: t38UDPRedundancy

```

Note that this feature does not add SDP attributes, but modifies the Capability Descriptor slightly. Payload type 127 is used for RFC 2833 support.

When the “Lucent RFC2833” switch is enabled, and all other switches are set to their default values, the SDP becomes:

```

v=0
o=- 382318343 382318343 IN IP4 10.1.36.219
s=-
c=IN IP4 10.1.36.219
t=0 0
m=audio 49688 RTP/AVP 0
b=AS: 81
a=rtcp-xr: voip-metrics
a=mptime: 20
a=rtpmap: 0 PCMU/8000/1
a=X- pc- secret: base64: Kue6n+ZSGpXrB2i AJI AUQNst6AtAS7Ad7zC3oGP
ry9XKdURiy4Y6i LaDEhk5lg== GgaMtuqF7/egj ksBpQ8SZeWnXCAI r1EeNH
AHV7EYOfv03Y0MYAYa1zz/Iv05dg==
a=X- pc- csuites- rtp: 62/51 64/51 60/51 60/50
a=X- pc- csuites- rtcp: 81/70 81/71 82/70 82/71 80/70
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 8 15 18 96 97 98 2 99 94
a=cpar: a=fmtp: 18 annexb=no
a=cpar: a=rtpmap: 94 telephone-event/8000/1
a=cpar: a=fmtp: 94 0-15, 144, 149, 159
a=cdsc: 11 image udptl t38
a=cpar: a=T38FaxVersion: 0
a=cpar: a=T38FaxRateManagement: transferredTCF
a=cpar: a=T38FaxMaxDatagram: 161
a=cpar: a=T38FaxUdpEC: t38UDPRedundancy

```

This feature does not add SDP attributes, but modifies the Capability Descriptor. In this case, payload type 94 is used for RFC 2833 support.



Note: The three feature switches that affect RFC 2833 negotiation are mutually exclusive. At most, only one of the bits may be set in the CallP Feature Switch. Enabling multiple RFC 2833 features may result in unexpected behavior.

When the “RFC2833 Digits” switch (**0x01000000**) is enabled, and all other switches are set to their default values, the SDP becomes:

```

v=0
o=- 382360201 382360201 IN IP4 10.1.36.219
s=-
c=IN IP4 10.1.36.219
t=0 0
m=audio 53560 RTP/AVP 0 101
b=AS: 81
a=rtcp-xr: voip-metrics
a=mptime: 20 -
a=rtpmap: 0 PCMU/8000/1
a=rtpmap: 101 telephone-event/8000/1
a=fmtp: 101 0-15
a=X- pc- secret: base64: rwZISK4HN5wl Zzehi 0BSEJXsRQbexmi wm10u4pE
nXFr4l STXQYdAsKFT5l hkkw== tWyPwAvBg6EbSs4+FoY7rW0n0l 8pcQPxGm
iwl NGPfo3Suehu0CncQ2egC4JQ6w==

```



```

a=X-pc-csuites-rtp: 62/51 64/51 60/51 60/50
a=X-pc-csuites-rtcp: 81/70 81/71 82/70 82/71 80/70
a=sgn: 0
a=cdsc: 1 audio RTP/AVP 0 8 15 18 96 97 98 2 99 101
a=cpar: a=fmtp: 18 annexb=no
a=cpar: a=fmtp: 101 0-15, 144, 149, 159
a=cdsc: 11 image udptl t38
a=cpar: a=T38FaxVersion: 0
a=cpar: a=T38FaxRateManagement: transferredTCF
a=cpar: a=T38FaxMaxDatagram: 161
a=cpar: a=T38FaxUdpEC: t38UDPRedundancy

```

Note that this feature affects the SDP, including the Capability Descriptor. the payload type specified by **ppCfgRfc2833DigitPayloadType** is added to the **m=** line (the default value is "101") and two new attributes indicate that the endpoint is prepared to receive RFC 2833 digits.

Secondary CallP Feature Switch

The secondary CallP feature switch, **ppCfgMtaCallPfeatureSwitch2**, provides more options for interoperability.

The following table lists the secondary CallP Feature Switch bits supported in AR01.1. The default value is **0x0**.

Bit Value	Description
0x00000001	(NCS only) Set this bit to treat OSI as busy.
0x00000002	Set this bit to ignore handling TDD tones.
0x00000004	<p>(NCS only) Set this bit to remove the optional "number of channels" encoding from the CODEC list in the SDP. For example, with this bit cleared, a typical SDP CODEC entry looks like:</p> <pre>8 PCMA/8000/1</pre> <p>With this bit set, the entry looks like:</p> <pre>8 PCMA/8000</pre>

Provisioning General eDVA and Line Parameters

Touchstone firmware provides a great deal of flexibility in configuring the eDVA and individual lines. This procedure covers general and miscellaneous parameters; other procedures in this chapter provide detailed information about more complex features.

Action

Perform the following tasks as needed.

- [Setting Persistent Line Status](#) 49
- [Controlling ToS Byte Marking](#) 49
- [Controlling TurboDOX Functionality](#) 50
- [Controlling IPsec Functionality](#) 50
- [Configuring the Ringing Waveform](#) 50
- [Configuring Loop Current](#) 51

Setting Persistent Line Status

Follow these steps to set persistent line status.

1. To set line status using SNMP, set the **arrisMtaDevPersistentLineStatus** object to either **ignore(0)** or **forceDisable(1)**.



Note: You must set this object using an SNMP manager. If this object is included in a configuration file, the eDVA ignores the setting.

Controlling ToS Byte Marking

By default, Touchstone firmware sets the ToS byte in RTCP packets to a value of 0. Follow these steps to change the value of the RTCP ToS byte.

1. Using an SNMP network manager, change the **arrisMtaDevRtcpTosValue** object to the desired value. Valid range: **0** to **63**.

The ToS marking changes immediately for all RTCP packets.



Note 1: The value of this object is shifted two bits before being loaded into the ToS byte.



Note 2: A common setting for ToS is **46** (expedited forwarding).

2. To make the change permanent, add the **arrisMtaDevRtcpTosValue** MIB object (with the desired value) to the eDVA provisioning file.

Controlling TurboDOX Functionality

TurboDOX is a TI-proprietary protocol, supported in hardware. Enabling TurboDOX increases performance of TCP-based protocols such as FTP and HTTP.

TurboDOX is enabled by default. You may need to disable TurboDOX during customer traffic testing and lab evaluations. ARRIS recommends that TurboDOX remained enabled in field deployments.

To provision TurboDOX functionality, add the **arrisCmDoc30SetupTurboDoxEnable** object to the CM configuration file. Set this object to **true**(1) to enable TurboDOX (the default), or **false**(2) to disable it.

Controlling IPsec Functionality

IPsec (Internet Protocol Security) is a collection of Internet standards used to encrypt and authenticate IP packets, to provide message integrity and privacy. IPsec provides security at the network layer (all TCP and UDP packets, and layers above).



Note: Touchstone E-UEs use only the IPsec ESP transport mode.

IPsec is enabled by default. Follow these steps to provision IPsec functionality.

1. For each CMS that the eDVA can communicate with, set the **pktcMtaDevCmsIpsecCtrl** object in the eDVA configuration file. The object is indexed by the CMS FQDN. Set the object to **true**(1) to enable IPsec between the eDVA and a particular CMS, and **false**(2) to disable it.
2. Reboot the eDVA to allow the new provisioning to take effect.

Configuring the Ringing Waveform

Telephony Modems support both trapezoidal and sinusoidal ringing waveforms when used with North American country templates. You may need to change the waveform if certain non-EIA compliant subscriber equipment does not recognize the trapezoidal waveform (in short, the phone does not ring).

The following defaults apply:

- North American templates: sinusoidal ringing
- All other country code templates: trapezoidal ringing



Note: Non-North American country code templates using other than 20 Hz ringing do not support sinusoidal ringing.

Configure the ringing waveform as follows.

**CAUTION****Potentially service-affecting**

Sinusoidal ringing is only supported for North American templates, and other templates using 20 Hz ringing. Use of this feature with other templates may result in loss of service.

1. Set the **arrisMtaDevEndPntRingingWaveform** object using one of the following methods:
 - a. In the eDVA configuration file, add the **arrisMtaDevEndPntRingingWaveform** object and set it to **sinusoidal** or **normal** as desired. Then reset the eDVA to allow the change to take effect.
 - b. Use an SNMP manager to access the eDVA and set the **arrisMtaDevEndPntRingingWaveform** object to **sinusoidal** or **normal** as desired.



Note: Setting this object in the configuration file does not write the setting to NVRAM; therefore, if you remove the object from the configuration file and reset the eDVA, it reverts to using the default waveform. Setting the object through SNMP does write the setting to NVRAM.

Configuring Loop Current

Touchstone Telephony Modems support a “boost,” or high loop current mode to compensate for faulty CPE or wiring. Use this setting, for example, to alleviate issues related to off-hook not being detected or fax machines failing to operate properly due to increased current draw by the equipment.

Loop current varies depending on boost mode and country template:

- Normal mode: 23 mA
- Boost mode (North American templates): 40 mA
- Boost mode (other templates): 33 mA

eDVAs using North American country templates (that is, the value of the **ppCfgMtaCountryTemplate** object is one of **northAmerica57(1)**, **northAmerica33(17)**, **northAmerica09(18)**, or **northAmerica66(32)**) default to high (boost) loop current. Other country templates default to normal loop current.



Note: If you set the loop current through SNMP, the eDVA does not retain the setting over reboots. If you set the loop current in the configuration file, the setting is written to NVRAM and retained over reboots.

1. To specify normal loop current (the default for non-North American loads), set the **ppCfgPortLoopCurrent.Line** object to **1**, where *Line* is the line number to set (beginning with **.1**).
2. To specify high loop current (the default for North American loads), set the **ppCfgPortLoopCurrent.Line** object to **2**.



Note: High loop current reduces battery hold-up times.

Configuring Caller ID Options

Follow these steps to configure optional Caller ID behavior.

1. Set the **arrisMtaDevDefaultReasonNoCIDName** object to specify what data the eDVA sends to the CPE device when the CID signal request includes no Caller ID name. Set the MIB object either directly through a network manager, or in the eDVA configuration file. Possible values are:

Value	Sends	Description
unavailable(0)	'O'	Caller ID displays typically show "Out of area" or "Unavailable." This is the default for all country templates except Switzerland.
private(1)	'P'	Caller ID displays typically show "Private."
sendnothing(2)	'00'	Sends NULL data to the CPE device.
sdmf(3)	number	Sends the number in NA SDMF format.
excludeName(4)	(blank)	Omits name parameters or any reason for the missing name.

2. To set the delay time, in milliseconds, between receiving the ACK from the CPE and transmitting the FSK, set the **arrisMtaDevOffHookFskDelay** object to the desired time. Valid range: **0** to **500** milliseconds. Default: **261** for North America loads, **100** (T12 timer value) for Euro-PacketCable loads.

Setting Loop Voltage Management

Touchstone firmware supports Loop Voltage Management as defined in PacketCable specification PKT-SP-MIB-EXSIG-I03-070412. Loop Voltage Management provides four management policies used to control loop voltage behavior during outages or cable cuts.

The Advanced "Product Details" web page shows the current loop voltage management settings under the "Optional Features" heading. If Policy 3 is set, the web page displays the reset timer setting.



Note: Touchstone firmware supports both PacketCable and ARRIS-proprietary MIB objects for provisioning Loop Voltage Management. See "see "[Mapping ARRIS Loop Voltage Objects to PacketCable Objects](#) (page 55)" for details.

Loop Voltage Management Policies

Use the **pktcEnNcsEndPntLVMgmtPolicy** to set the Loop Voltage Management policy. The following policies are supported:

- **vol t ageAtAl lTi mes**(1)
- **vol t ageUnl essRFQAMabsent**(2)
- **vol t ageBasedOnServi ceOrTi mers**(3)
- **vol t ageBasedOnServi ce**(4) (default)

Policy 1: Constant Loop Voltage

When this option is selected, the eDVA maintains loop voltage at all times with the following two exceptions:

- During firmware initialization of the line card. Touchstone eDVAs remove loop voltage for up to 1 second during firmware initialization, although typically this time is shorter.
- When the unit has no power.

Policy 2: QAM Carrier Detect

When this option is selected, the eDVA maintains loop voltage when it can lock onto a QAM carrier, including digital video QAM carriers. The assumption is that if the Telephony Modem can recognize a carrier, the connection is intact (has not been cut by a burglar).

When the Telephony Modem loses its carrier, after the T4 timeout expires (20–30 seconds), the modem scans cached and preset frequencies, then scans the entire spectrum. If the cable is truly cut, the scan takes 1 to 2 minutes to complete. If the modem lost its carrier, but can detect other RF energy (such as analog carriers), the scan can take up to 30 minutes to complete. If the modem does not detect any QAM carriers after scanning the STD and LRC frequencies, it removes loop voltage then continues with slow scanning.

Once the Telephony Modem removes loop voltage, it does not re-apply loop voltage until it re-registers with eDVA provisioning.

Policy 3: eDVA In-Service/Manual Reset

When this option is selected, the Telephony Modem maintains loop voltage when in-service or during manually-initiated resets and T4 timeouts. The Telephony Modem is considered in-service when eDVA TFTP is complete. Both subscriber resets (pushing the Reset button) and headend-initiated resets (SNMP, firmware upgrade) are considered manually-initiated.

Two timers govern the behavior of the eDVA during resets or outages:

pktcEnNcsEndPntLVMgmtResetTimer

The reset timer determines how long the eDVA maintains loop voltage during a reset. The default value is **3** for .TW loads, and **5** in all other loads.



Note: If the reset timer is set to a period longer than the scanning time between resets (typically 7 minutes), the eDVA never drops loop voltage.

pktcEnNcsEndPntLVMgmtMaintTimer

The plant maintenance timer (PMT) determines how long the eDVA maintains loop voltage when the E-UE loses its downstream signal. This timer may be used to maintain loop voltage during extended plant maintenance intervals.



Note: The value returns to the default setting, following a Telephony Modem reset.

When either timer expires without the eDVA coming in-service, the eDVA drops loop voltage.

Policy 4: eDVA In-Service

This is the default policy. Using this option, the eDVA goes through the following steps:

1. When applying initial AC power, Telephony Modems do not apply loop voltage.
2. When the eDVA completes TFTP and the lines are provisioned, it applies loop voltage only if the eDVA completes provisioning and the **pktcMtaDevProvisioningState** MIB object has one of the following values:
 - **pass(1)**
 - **passWithWarnings(4)**
 - **passWithIncompleteParsing(5)**

If provisioning succeeds, the eDVA applies loop voltage to provisioned lines. If a provisioned line goes off-hook before the eDVA has contacted the call server, the modem immediately attempts to contact the call server and allows the call to continue. Loop voltage is applied even if the call server cannot be reached.

3. After the eDVA device is in service and there is an interruption to the RF, loop voltage remains present on the lines until a T4 timeout occurs (generally 20 to 30 seconds).

Loop Voltage Management MIB Objects

The **pktcEnNcsEndPntLVMgmtTable** table contains the loop voltage policy and timers. This table is indexed by **ifIndex**; an index value of **1** applies the policy settings to all eDVA lines.

The objects in the **pktcEnNcsEndPntLVMgmtTable** are:

pktcEnNcsEndPntLVMgmtPolicy

Defines the policy; one of:

- **voltageAtAllTimes(1)**
- **voltageUnlessRFQMAbsent(2)**
- **voltageBasedOnServiceOrTimers(3)**
- **voltageBasedOnService(4)**

See ["Loop Voltage Management Policies"](#) (page 53) for descriptions of each policy.

pktcEnNcsEndPntLVMgmtResetTimer

The time, in minutes, allowed for an eDVA to successfully provision after a reset. This timer applies only when **pktcEnNcsEndPntLVMgmtPolicy** is set to a value of **vol tageBasedOnServi ceOrTi mers**(3). In all other cases, reading this object returns a value of zero.

The eDVA starts the timer upon a hard reboot, a soft reset or a T4 timeout. The timer value persists the last configured value (i.e., not the countdown value) of this MIB Object across hard reboots and soft resets.

Valid range: **0** to **1440** (minutes). Default: **5**.

pktcEnNcsEndPntLVMgmtMaintTimer

The time, in minutes, that the eDVA maintains loop voltage regardless of the eDVA's connection or provisioning status. This timer applies only when **pktcEnNcsEndPntLVMgmtPolicy** is set to a value of **vol tageBasedOnServi ceOrTi mers**(3). In all other cases, reading this object returns a value of zero. The current timer value persists across soft resets, but resets to zero for a hard reset or power-cycle.

The eDVA starts the timer when it is set to a value greater than zero. The eDVA maintains loop voltage until the timer expires.

Valid range: **0** to **1440** (minutes). Default: **0**.

Mapping ARRIS Loop Voltage Objects to PacketCable Objects

Some earlier versions of Touchstone firmware provided an ARRIS-proprietary version of Loop Voltage Management that is very similar to the PacketCable-standard version. For backward compatibility, AR01.1 supports both ARRIS and PacketCable MIB objects.

The following table defines how ARRIS objects map to PacketCable objects.

ARRIS Object	PacketCable Object	Notes
arrisMtaDevLoopVoltageKey	(none)	Not needed
arrisMtaDevLoopVoltagePolicy al ways- vol tage- present (1) rf- carri er- vol tage- present (2) i n- servi ce- vol tage- present (3) defaul t- operati on (4)	pktcEnNcsEndPntLVMgmtPolicy vol tageAtAl l Ti mes (1) vol tageUnl essRFQAMAbsent (2) vol tageBasedOnServi ceOrTi mers (3) vol tageBasedOnServi ce (4)	Roughly equivalent
arrisMtaDevLoopVoltageResetTimeout ■ Valid range: 8–1800	pktcEnNcsEndPntLVMgmtResetTimer ■ Valid range: 0–1400	Adjust ranges as shown
arrisMtaDevLoopVoltageMaintTimeout ■ Valid range: integer	pktcEnNcsEndPntLVMgmtMaintTimer ■ Valid range: 0–1440	Adjust ranges as shown

Action

Follow these steps to configure loop voltage management.

1. Set the **pktcEnNcsEndPntLVMgmtPolicy** object to the appropriate policy value:

- **vol t ageAtAl l Ti mes**(1)
- **vol t ageUnl essRFQAMabsent**(2)
- **vol t ageBasedOnServi ceOrTi mers**(3)
- **vol t ageBasedOnServi ce**(4) (default)



Note: For this and other LVM objects, use an index of **. 1** to apply the same policy to all lines. If you want to specify different policies for each line, use the **ifIndex** of each line instead.

2. For eDVAs using policy 3, modify the reset timer (if necessary) by setting the **pktcEnNcsEndPntLVMgmtResetTimer** object. Valid range: **0** to **1440** minutes. Default: **5** minutes.

The line card drops loop voltage after a CM reset, if the eDVA has not been successfully provisioned before the Reset Timeout timer expires.



Note: This object is only used with policy option 3 and is ignored if a policy setting of other than 3 is used.

3. For eDVAs using policy 3, modify the plant maintenance timer (if necessary) by setting the **pktcEnNcsEndPntLVMgmtMaintTimer** object. Valid range: **0** to **1440** minutes (24 hours).
4. Reset the eDVA to enable the new loop voltage policy on the eDVA.

Echo Cancellation and Analog Fax/Modem Support

Touchstone eDVAs support:

- echo cancellation per ITU G.168/G, with a 28 dB echo return loss
- Group I-III compliant facsimile devices
- analog voice band modems up to v.92

Echo cancellation provides a 32 ms echo cancellation tail.

Echo cancellation operates in one of three modes:

- enabled
- enabled with non-linear processor (NLP) disabled
- disabled

The echo cancellation feature is always enabled for normal voice calls. However, for analog fax/modem calls, the ARRIS eDVA sets the echo cancellation mode depending on detected tones:

Tones Detected	Mode
Calling (CNG), V.21 preamble, or “slow” speed CED (14.4k or lower)	NLP disabled
“high” speed CED (28.8k and higher)	completely disabled

If the eDVA detects fax/modem tones, and a CODEC other than G.711 is active, the eDVA automatically switches to the G.711 CODEC if G.711 was negotiated as a backup CODEC when the call was set up. Upon completion of a fax call, the eDVA automatically re-enables echo cancellation, but does not switch back to the original CODEC unless instructed to switch by the Call Agent. After a modem call completes, the eDVA re-enables echo cancellation when instructed by the Call Agent.

Adaptive Jitter Buffers

Touchstone firmware supports jitter buffers for fax/modem calls. These buffers automatically adapt their size to accommodate the required jitter. See [Configuring Jitter Buffers](#) (page 67) for configuring jitter buffer sizes.

Configuring the Echo Cancellation Tail Length

The default echo cancellation tail length is 32 ms. Follow these steps to configure the desired echo cancellation tail length.

1. In the eDVA configuration file, add the **arrisMtaDevEchoCancellerTailLength** object and set it to **eightMs** or **thirtyTwoMs** (default) as desired.
2. Reset the eDVA.

The eDVA sets the echo cancellation tail as configured, then marks the MIB read-only.

Provisioning RFC 2833 Support

RFC 2833 (also called DTMF Relay) specifies a method for carrying DTMF and other telephony signals and events in RTP packets, instead of sending audio tones over the network. This functionality is especially important when using highly-compressed CODECs such as G.729, which may distort DTMF tones.

The eDVA signals RFC 2833 support by specifying “telephone-event” in its list of available CODECs during negotiation. By default, the CMS instructs the eDVA to enable or disable RFC 2833 and selects the payload type to use. Touchstone firmware provides feature switches to override the CMS and enable RFC 2833 support with a specific payload type.

Controlling RFC 2833 Functionality

Touchstone firmware can enable RFC 2833 functionality regardless of whether the CMS instructs the eDVA to use it. Two MIB objects control the functionality:

ppCfgMtaCallPFeatureSwitch

The following CallP Feature Switch bits control RFC 2833 functionality:

Bit	Description
0x00001000	Nuera RFC 2833 messaging without request using payload 127 (NCS only) Set this bit to instruct the eDVA to generate RFC 2833 events with a payload type of 127 without specifically being instructed to do so (for compatibility with the Nuera RDT).
0x00080000	LUCENT RFC 2833 messaging without request using payload 94 (NCS only) Set this bit to instruct the eDVA to generate RFC 2833 events with a payload type of 94 during call setup, without specifically being instructed to do so.
0x01000000	Send DTMF digits via RFC 2833 with operator-specified payload without request (NCS only) Set this bit to instruct the eDVA to generate RFC 2833 DTMF events using a payload type defined in the ppCfgRfc2833DigitPayloadType MIB object without being instructed to do so by the CMS. The default payload type value is 101.

ppCfgRfc2833DigitPayloadType

When bit 0x01000000 is enabled in the CallP Feature Switch, the eDVA sends RFC 2833 events with the payload type specified by this object.

Valid range: **97** to **127**. Default: **101**.

Configuring T.38 Fax Relay Support

Touchstone firmware supports T.38 fax relay, version 0. T.38 fax relay provides higher reliability of fax transmissions using redundancy to tolerate packet loss. Touchstone firmware supports call agent-controlled T.38 as defined in PKT-SP-NCS1.5-I03-070412, Appendix A.

T.38 support requires that SDP capability reporting be enabled (the default setting).

SDP Parameter List for T.38 Strict

When T.38 Strict mode is enabled, the eDVA sends an SDP list as shown below. The bolded portion indicates the capability descriptor.

```
v=0
o=- 48186 48188 IN IP4 10.1.36.218
s=-
c=IN IP4 10.1.36.218
t=0 0
```

```

m=audio 61304 RTP/AVP 0 8 101
a=rtpmap: 0 PCMU/8000
a=rtpmap: 8 PCMA/8000
a=rtpmap: 101 telephone-event/8000
a=fmtp: 101 0-15
a=sendrecv
a=ptime: 20
a=sn: 0
a=cdsc: 1 audio RTP/AVP 0 8 15 18 96 97 98 2 99 101
a=cpar: a=rtpmap: 96 G729E/8000/1
a=cpar: a=rtpmap: 97 G726-16/8000/1
a=cpar: a=rtpmap: 98 G726-24/8000/1
a=cpar: a=rtpmap: 2 G726-32/8000/1
a=cpar: a=rtpmap: 99 G726-40/8000/1
a=cdsc: 11 image udptl t38
a=cpar: a=T38FaxVersion: 0
a=cpar: a=T38FaxRateManagement: transferredTCF
a=cpar: a=T38FaxMaxDatagram: 160
a=cpar: a=T38FaxUdpEC: t38UDPRedundancy
a=cpar: a=T38MaxBitRate: 14400
m=image 0 udptl t38

```

If the capability descriptor causes interoperability issues, set bit 0x00000008 of the CallP Feature Switch. See [CallP Feature Switch](#) (page 40) for more information about the feature switch.

CallP Feature Switches Affecting the SDP

The following feature switches affect the SDP, returned in response to a Create (CRCX) or Modify (MDCX) Connection command.

Bit	Description
0x00000008	Reduce the capability descriptor in the SDP to T38 only (default = 0, no reduction).
0x00000080	Omit mptime parameter in returned SDP (default = 0, mptime included).
0x00001000	NUERA RFC 2833 messaging without request using payload 127 (default = 0, telephone-event is negotiated normally).
0x00080000	LUCENT RFC 2833 messaging without request using payload 94 (default = 0, telephone-event is negotiated normally).
0x00100000	Allow AES encryption for RTP/RTCP (default = 1, AES encryption is negotiated normally).
0x01000000	Send DTMF digits via RFC 2833 with an operator-defined payload without request (default = 0, telephone-event is negotiated normally).

The following CRCX message is used to generate all the SDP examples, unless otherwise specified:

```

CRCX 19901 aal n/1@mta218.dev36 MGCP 1.0 NCS 1.0
C: 1234
M: recvonly
L: mp: 20, a: PCMU, fxr/fx: t38-loose, xrm/mcr: on

```

The default feature switch settings are (**0x0** for NCS loads, and **0x0020** for SIP loads) generate the following SDP:

```

v=0
o=- 381749076 381749076 IN IP4 10.1.36.219
s=-
c=IN IP4 10.1.36.219
t=0 0
m=audio 65496 RTP/AVP 0
b=AS: 81
a=rtcp-xr: voip-metrics
a=mptime: 20
a=rtpmap: 0 PCMU/8000/1
a=X-pc-secret: base64: ZNos/qs530eSDJ4FvdL2GJBR62lS5UKyQ7n9og4
IaadbA9Blpg6lM2Pf0aHEGg== U5Q5N/eWhi mq9Q/yj WwY2hACRI Y6a9qqEQ
Us8tm54lEmEE6LXkCB51+3sqxlQg==
a=X-pc-csuites-rtp: 62/51 64/51 60/51 60/50
a=X-pc-csuites-rtcp: 81/70 81/71 82/70 82/71 80/70
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 8 15 18 96 97 98 2 99 101
a=cpar: a=fmtp: 18 annexb=no
a=cpar: a=rtpmap: 101 telephone-event/8000/1
a=cpar: a=fmtp: 101 0-15, 144, 149, 159
a=cdsc: 11 image udptl t38
a=cpar: a=T38FaxVersion: 0
a=cpar: a=T38FaxRateManagement: transferredTCF
a=cpar: a=T38FaxMaxDatagram: 161
a=cpar: a=T38FaxUdpEC: t38UDPRedundancy

```

When the “Reduce Capability Descriptor” switch (**0x00000008**) is enabled, and all other switches are set to their default values, the SDP becomes:

```

v=0
o=- 381749076 381749076 IN IP4 10.1.36.219
s=-
c=IN IP4 10.1.36.219
t=0 0
m=audio 65496 RTP/AVP 0
b=AS: 81
a=rtcp-xr: voip-metrics
a=mptime: 20
a=rtpmap: 0 PCMU/8000/1
a=X-pc-secret: base64: ZNos/qs530eSDJ4FvdL2GJBR62lS5UKyQ7n9og4
IaadbA9Blpg6lM2Pf0aHEGg== U5Q5N/eWhi mq9Q/yj WwY2hACRI Y6a9qqEQ
Us8tm54lEmEE6LXkCB51+3sqxlQg==
a=X-pc-csuites-rtp: 62/51 64/51 60/51 60/50
a=X-pc-csuites-rtcp: 81/70 81/71 82/70 82/71 80/70
a=sqn: 0
a=cdsc: 1 image udptl t38

```

In this example, the Capability Descriptor is reduced to only the information required to relay support for UDPTL T.38 to the far end. This allows the far end to support T.38 strict mode as defined in the PacketCable 1.5 NCS specification.

When the “Omit mptime” switch (**0x00000080**) is enabled, and all other switches are set to their default values, the SDP becomes:

```

v=0
o=- 382093395 382093395 IN IP4 10.1.36.219

```

```

S=-
c=IN IP4 10.1.36.219
t=0 0
m=audio 58810 RTP/AVP 0
b=AS: 81
a=rtcp-xr: voip-metrics
a=rtpmap: 0 PCMU/8000/1
a=X-pc-secret: base64: XI51bgXR5MNdaKXi sS0tj YCc90x3f7j A+oj yam
W/0/M2BI CaejlrRL0dApR6w== 6LbN8ULCFGMj cR2T3l 1uZuBcfWM2vfGn09
YTmR60hHfQwC4eE+WMSX7AarnFPA==
a=X-pc-suites-rtp: 62/51 64/51 60/51 60/50
a=X-pc-suites-rtcp: 81/70 81/71 82/70 82/71 80/70
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 8 15 18 96 97 98 2 99 101
a=cpar: a=fmtp: 18 annexb=no
a=cpar: a=rtpmap: 101 telephone-event/8000/1
a=cpar: a=fmtp: 101 0-15, 144, 149, 159
a=cdsc: 11 image udptl t38
a=cpar: a=T38FaxVersion: 0
a=cpar: a=T38FaxRateManagement: transferredTCF
a=cpar: a=T38FaxMaxDatagram: 161
a=cpar: a=T38FaxUdpEC: t38UDPRedundancy

```

In this example, neither the `a=ptime` nor the `a=ptime` parameters are included in the SDP since the packetization rate is the default (20 ms). If the Call Agent had specified a different packetization rate, then the `a=ptime` parameter is included as follows:

```

v=0
o=- 382093395 382093395 IN IP4 10.1.36.219
S=-
c=IN IP4 10.1.36.219
t=0 0
m=audio 58810 RTP/AVP 0
b=AS: 81
a=rtcp-xr: voip-metrics
a=ptime: 10
a=rtpmap: 0 PCMU/8000/1
a=X-pc-secret: base64: XI51bgXR5MNdaKXi sS0tj YCc90x3f7j A+oj yam
W/0/M2BI CaejlrRL0dApR6w== 6LbN8ULCFGMj cR2T3l 1uZuBcfWM2vfGn09
YTmR60hHfQwC4eE+WMSX7AarnFPA==
a=X-pc-suites-rtp: 62/51 64/51 60/51 60/50
a=X-pc-suites-rtcp: 81/70 81/71 82/70 82/71 80/70
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 8 15 18 96 97 98 2 99 101
a=cpar: a=fmtp: 18 annexb=no
a=cpar: a=rtpmap: 101 telephone-event/8000/1
a=cpar: a=fmtp: 101 0-15, 144, 149, 159
a=cdsc: 11 image udptl t38
a=cpar: a=T38FaxVersion: 0
a=cpar: a=T38FaxRateManagement: transferredTCF
a=cpar: a=T38FaxMaxDatagram: 161
a=cpar: a=T38FaxUdpEC: t38UDPRedundancy

```

When the “Allow AES Encryption” switch (**0x00100000**) is disabled, and all other switches are set to their default values, the SDP becomes;

```

v=0
o=- 381749076 381749076 IN IP4 10.1.36.219
S=-
c=IN IP4 10.1.36.219
t=0 0
m=audio 65496 RTP/AVP 0
b=AS: 81

```

```

a=rtcp-xr: voip-metrics
a=mptime: 20
a=rtpmap: 0 PCMU/8000/1
a=sn: 0
a=cdsc: 1 audio RTP/AVP 0 8 15 18 96 97 98 2 99 101
a=cpar: a=fmtp: 18 annexb=no
a=cpar: a=rtpmap: 101 telephone-event/8000/1
a=cpar: a=fmtp: 101 0-15, 144, 149, 159
a=cdsc: 11 image udptl t38
a=cpar: a=T38FaxVersion: 0
a=cpar: a=T38FaxRateManagement: transferredTCF
a=cpar: a=T38FaxMaxDatagram: 161
a=cpar: a=T38FaxUdpEC: t38UDPRedundancy

```

The last three CPFS bits (**0x00001000**, **0x00080000**, and **0x01000000**) are related to RFC 2833. They are used in specific configurations and are designed to skip CODEC negotiation. For example, when the “Nuera RFC2833” feature is enabled, and all other switches are set to their default values, the SDP becomes:

```

v=0
o=- 382250430 382250430 IN IP4 10.1.36.219
s=-
c=IN IP4 10.1.36.219
t=0 0
m=audio 63672 RTP/AVP 0
b=AS: 81
a=rtcp-xr: voip-metrics
a=mptime: 20
a=rtpmap: 0 PCMU/8000/1
a=X- pc- secret: base64: Qb8GFLNXP4Z3yi yxFx1Ws9vLph9qG6bTIXezUl z
rwi a7i i NvPPkVYdZhZ77NEQ== zZj gwXRR2j 5F04l DXefPTV06PT8g31Hn5V
Ea6NJvFPFsPi raDeDI 35EI 8K0+4A==
a=X- pc- csuites- rtp: 62/51 64/51 60/51 60/50
a=X- pc- csuites- rtcp: 81/70 81/71 82/70 82/71 80/70
a=sn: 0
a=cdsc: 1 audio RTP/AVP 0 8 15 18 96 97 98 2 99 127
a=cpar: a=fmtp: 18 annexb=no
a=cpar: a=rtpmap: 127 telephone-event/8000/1
a=cpar: a=fmtp: 127 0-15, 144, 149, 159
a=cdsc: 11 image udptl t38
a=cpar: a=T38FaxVersion: 0
a=cpar: a=T38FaxRateManagement: transferredTCF
a=cpar: a=T38FaxMaxDatagram: 161
a=cpar: a=T38FaxUdpEC: t38UDPRedundancy

```

Note that this feature does not add SDP attributes, but modifies the Capability Descriptor slightly. Payload type 127 is used for RFC 2833 support.

When the “Lucent RFC2833” switch is enabled, and all other switches are set to their default values, the SDP becomes:

```

v=0
o=- 382318343 382318343 IN IP4 10.1.36.219
s=-
c=IN IP4 10.1.36.219
t=0 0
m=audio 49688 RTP/AVP 0
b=AS: 81
a=rtcp-xr: voip-metrics
a=mptime: 20
a=rtpmap: 0 PCMU/8000/1
a=X- pc- secret: base64: Kue6n+ZSGpXrB2i AJIAUQNst6AtAS7Ad7zC3oGP
ry9XKdURi y4Y6i LaDEhk5l g== GgaMt uqF7/egj ksBpQ8SZeWnXCAI r1EeNH
AHV7EYOfv03Y0MYAYa1zz/lv05dg==

```

```

a=X-pc-csuites-rtp: 62/51 64/51 60/51 60/50
a=X-pc-csuites-rtcp: 81/70 81/71 82/70 82/71 80/70
a=sn: 0
a=cdsc: 1 audio RTP/AVP 0 8 15 18 96 97 98 2 99 94
a=cpar: a=fmtp: 18 annexb=no
a=cpar: a=rtpmap: 94 telephone-event/8000/1
a=cpar: a=fmtp: 94 0-15, 144, 149, 159
a=cdsc: 11 image udptl t38
a=cpar: a=T38FaxVersion: 0
a=cpar: a=T38FaxRateManagement: transferredTCF
a=cpar: a=T38FaxMaxDatagram: 161
a=cpar: a=T38FaxUdpEC: t38UDPRedundancy

```

This feature does not add SDP attributes, but modifies the Capability Descriptor. In this case, payload type 94 is used for RFC 2833 support.



Note: The three feature switches that affect RFC 2833 negotiation are mutually exclusive. At most, only one of the bits may be set in the CallP Feature Switch. Enabling multiple RFC 2833 features may result in unexpected behavior.

When the “RFC2833 Digits” switch (**0x01000000**) is enabled, and all other switches are set to their default values, the SDP becomes:

```

v=0
o=- 382360201 382360201 IN IP4 10.1.36.219
s=-
c=IN IP4 10.1.36.219
t=0 0
m=audio 53560 RTP/AVP 0 101
b=AS: 81
a=rtcp-xr: voip-metrics
a=mptime: 20 -
a=rtpmap: 0 PCMU/8000/1
a=rtpmap: 101 telephone-event/8000/1
a=fmtp: 101 0-15
a=X-pc-secret: base64: rwZISK4HN5wl ZzeHi 0BSEJXsRQbexmi wm10u4pE
nXFr4l STXQYdAsKFT5l hkkw== tWyPwAvBg6EbSs4+FoY7rWOn0l 8pcQPXGm
iwl NGPfo3Suehu0CncQ2egC4JQ6w==
a=X-pc-csuites-rtp: 62/51 64/51 60/51 60/50
a=X-pc-csuites-rtcp: 81/70 81/71 82/70 82/71 80/70
a=sn: 0
a=cdsc: 1 audio RTP/AVP 0 8 15 18 96 97 98 2 99 101
a=cpar: a=fmtp: 18 annexb=no
a=cpar: a=fmtp: 101 0-15, 144, 149, 159
a=cdsc: 11 image udptl t38
a=cpar: a=T38FaxVersion: 0
a=cpar: a=T38FaxRateManagement: transferredTCF
a=cpar: a=T38FaxMaxDatagram: 161
a=cpar: a=T38FaxUdpEC: t38UDPRedundancy

```

Note that this feature affects the SDP, including the Capability Descriptor. the payload type specified by **ppCfgRfc2833DigitPayloadType** is added to the **m=** line (the default value is “101”) and two new attributes indicate that the endpoint is prepared to receive RFC 2833 digits.

PacketCable 1.5 Extended Signaling

Touchstone firmware supports the PacketCable 1.5 NCS Extended Signaling MIB fax detection objects for T.38 signaling. The following objects are supported:

pktcEnNcsMinimumDtmfPayout

The minimum time a digit is played when the eDVA receives an RFC 2833 digit event.

pktcEnNcsEndPntQuarantineState

The endpoint state, based on the NCS quarantine states (for example, notification or lockstep).

pktcEnNcsEndPntHookState

The hook state (on-hook or off-hook) of the endpoint.

pktcEnNcsEndPntFaxDetection

(NCS only) Configures whether the eDVA uses T.38 when detecting a CNG (calling) tone. The default is disabled, which prevents using T.38.

SDP Parameter List Considerations

The complete list of Call Processing features (including T.38 Fax Relay) requires a Session Description Protocol (SDP) parameter list longer than 512 bytes. Some CMSs or P-CSCFs do not support a parameter list longer than 512 bytes. To reduce the SDP parameter list size, set the “Suppress SDP Capability Attribute Parameters” CallP Feature Switch (bit **0x00000100**) to **1**.

Setting this bit disables T.38 strict mode functionality; T.38 loose mode is still available.

For more SDP-related options, see the CallP Feature Switch details in [CallP Feature Switch](#) (page 40).

T.38 Provisioning Overview

The following MIB objects control T.38 behavior.

sipCfgPortT38Mode

(SIP only) Sets the T.38 operating mode. It allows the following values:

Value	Description
t380ff(1)	(default) Disables T.38.
t38Loose(2)	Enables T.38 Loose mode. In loose mode, the eDVA can use T.38 for fax transmission whether or not the remote endpoint has indicated T.38 support.
t38Strict(3)	Enables T.38 Strict mode. In strict mode, the eDVA can use T.38 for fax transmission only if the far end indicated support for T.38 during session negotiation.

arrisMtaDevEndPntFaxOnlyLineTimeout

(NCS only) Configures fax-only mode for a line. Specifies the time, in seconds, to wait for fax or modem tones after receiving the SDP from the remote endpoint. If the time expires before detecting fax or modem tones, the eDVA drops the call. Valid range: **0** to **600**.

sipCfgPortFaxOnlyTimeout

(SIP only) Configures fax-only mode for a line. Specifies the time, in seconds, to wait for fax or modem tones after receiving the SDP from the remote endpoint. If the time expires before detecting fax or modem tones, the eDVA drops the call. Valid range: **0** to **600**.

sipCfgPortMaxT38HSRedLevel

(SIP only) Sets the maximum high-speed redundancy level used for T.38 fax relay, in both send and receive directions. The value of this object is the number of older data packets included in each T.38 datagram when transferring fax data. The actual redundancy level used is determined by negotiation with the remote endpoint. Valid range: **0** to **2**. Default: **1**.

arrisMtaDevT38Timeout

(NCS only) The time, in seconds, that the eDVA waits for the Call Agent to modify the connection to T.38. The far end receives silence until the Call Agent modifies the connection.

Valid range: **1** to **30** seconds. Default: **10**.

Action

Perform the following tasks as needed.

- [Controlling T.38 and Fax-Only Modes](#) 65
- [Configuring T.38 MaxDatagram Size](#) 66

Controlling T.38 and Fax-Only Modes

Follow these steps to configure T.38 fax detection and fax-only mode.

1. Set the **pktcEnNcsEndPntFaxDetection** object (use the line number as an index, starting with 1) to configure whether the eDVA detects CNG (calling) tones and starts T.38 mode. The default value, **false**(2), disables detection of CNG tones on the endpoint. Disabling CNG detection prevents the eDVA from using T.38 mode, which may be desired when the fax machines are capable of Super G3 (which uses a higher transmission rate).



Note: If you make this change using an SNMP browser, the new setting takes effect on the next connection.

2. If needed, set the **pktcEnNcsMinimumDtmfPlayout** object to specify the minimum time, in milliseconds, that the eDVA plays out a digit tone when receiving an RFC 2833 digit event on the specified endpoint. The actual play-out time is the maximum of this setting and the time specified in the RFC 2833 packet.

Valid range: **40** to **100**, or **0** (the default), which always uses the time specified in the RFC 2833 packet.

3. Set fax-only mode by adding the **arrisMtaDevEndPntFaxOnlyLineTimeout** MIB object to the eDVA configuration file. This MIB object is specific to a line, so it must be specified with the line number; for example, **arrisMtaDevEndPntFaxOnlyLineTimeout.1** for line 1. The value specifies the timeout, in seconds, after which the eDVA drops the call if it does not detect fax or modem tones.

Valid range (either object): **0** (disabled) to **600** seconds. The default is **0**.

Configuring T.38 MaxDatagram Size

Follow these steps to configure the MaxDatagram size for T.38.

1. In the eDVA configuration file, set the **ppCfgPortT38MaxDatagram** object to the desired datagram size.

Valid range: **160** to **65535** octets. Default: **160**.

2. Reset the eDVA to have the new datagram size take effect.

Super G3 FAX Support

Touchstone firmware supports SuperG3 FAX transmission, including V.8 data exchange detection.

When the eDVA detects the V.8 signal, fax transmission proceeds depending on the setting of the **arrisMtaDevSuperG3FaxRelay** MIB object:

- When set to **disable(0)** (the default), the eDVA uses the V.8 detection as a trigger to set up the endpoint for FAX transmission (adjust the jitter buffer, turn off echo cancellation) and allow the FAX machines to handle the negotiation and transmit the fax using G.711.
- When set to **enable(1)**, the eDVA uses the V.8 detection as a trigger to start T.38.

To switch the FAX call from SuperG3 to T.38, the Telephony Modem mutes audio to prevent the FAX machine from negotiating to SuperG3 before switching to T.38.

Depending on the T.38 negotiation, this switch either forces the FAX machines to downshift to G3 speeds (≤ 14.4) which is the current maximum rate supported for T.38 version 0, or initiate the T.38 over RTP.

If the connection is negotiated for T.38 version 0, the audio remains muted until the CMS transitions the connection to T.38. During the muted period, the terminating fax machine receives silence. The maximum amount of this silence period can be controlled by the **arrisMtaDevT38Timeout** MIB object. This object defaults to 30 seconds, which is consistent with existing behavior. Alter this value only if FAX transmissions are consistently failing due T.38 not re-trying.

Configuring Jitter Buffers

Touchstone firmware provides three sets of MIB objects to adjust the eDVA jitter buffer:

- **Standard**—used for typical voice calls. Associated MIB objects are:
 - The **arrisMtaDevVPJitterBufferMode** object controls the jitter buffer behavior (adaptive or fixed modes).
 - The **arrisMtaDevVPNomJitterBuffer** object sets the nominal jitter buffer size (in packets).
 - The **arrisMtaDevVPMajJitterBuffer** object sets the maximum jitter buffer size (in packets).
- **Voice Band Data (VBD)**—used for calls involving data transfer, including fax, modem, and POS terminals. The following MIB objects allow the eDVA to automatically override the normal jitter buffer settings when it detects a Voice Band Data call:
 - The **arrisMtaDevVbdOverwriteLineBitmap** object allows per-line control of jitter buffer override.
 - The **arrisMtaDevVbdOverwriteMinJitterBuffer** object specifies the minimum Voice Band Data call jitter buffer setting.
 - The **arrisMtaDevVbdOverwriteNomJitterBuffer** object specifies the nominal Voice Band Data call jitter buffer setting.
 - The **arrisMtaDevVbdOverwriteMaxJitterBuffer** object specifies the maximum Voice Band Data call jitter buffer setting.

If you change these parameters with an SNMP manager, the new settings take effect starting with the next phone call.

- **Custom**—provides more precise control over jitter buffer settings. Custom settings may be useful under certain network conditions or applications.
 - The **arrisMtaDevCustomJitterBufferEnabled** object enables customer jitter buffer settings.
 - The **arrisMtaDevCustomMinJitterBuffer** object controls the minimum custom jitter buffer duration.
 - The **arrisMtaDevCustomNomJitterBuffer** object controls the nominal custom jitter buffer duration.
 - The **arrisMtaDevCustomMaxJitterBuffer** object controls the maximum custom jitter buffer duration.

Action

Perform the following tasks as needed.

- [Setting Standard Jitter Buffer Parameters](#) 68
- [Setting Voice Band Data Jitter Buffer Parameters](#) 68
- [Configuring Custom Jitter Buffer Settings](#) 69

Setting Standard Jitter Buffer Parameters

Follow these steps to set standard jitter buffer parameters.

1. Set the **arrisMtaDevVPJitterBufferMode** object to either **1** (adaptive, the default) or **2** (fixed).
2. Configure the nominal voice call jitter buffer size by setting the **arrisMtaDevVPNomJitterBuffer** object. The value represents a multiple of the packetization rate. Valid range: **1** to **4**. Default: **1**.
3. Configure the maximum voice call jitter buffer size by setting the **arrisMtaDevVPMaxJitterBuffer** object. The value represents a multiple of the packetization rate. Valid range: **1** to **4**. Default: **3**.



Note: The nominal jitter buffer setting must be less than the maximum jitter buffer setting.

Setting Voice Band Data Jitter Buffer Parameters

Follow these steps to enable and configure jitter buffer and override settings for Voice Band Data (fax, modem, POS terminal) calls.

1. Set the **arrisMtaDevVbdOverwriteLineBitmap** to control jitter buffer override on each line. The least significant bit controls line 1, so a value of **3** enables override on lines 1 and 2.
2. Configure the minimum, nominal, and maximum jitter buffer settings for Voice Band Data calls by setting the following objects. The valid range for all three objects is **10** to **135**; the following table shows the default value for each object.

Object	Default Value	Description
arrisMtaDevVbdOverwriteMinJitterBuffer	25	Minimum fax/modem call jitter buffer setting.
arrisMtaDevVbdOverwriteNomJitterBuffer	25	Nominal fax/modem call jitter buffer setting.
arrisMtaDevVbdOverwriteMaxJitterBuffer	135	Maximum fax/modem call jitter buffer setting.



Note: The minimum jitter buffer setting must be less than the nominal setting, which in turn must be less than the maximum jitter buffer setting.

Configuring Custom Jitter Buffer Settings

Follow these steps to configure custom jitter buffer settings.



Note: Custom jitter buffer settings use units of milliseconds, rather than packet multiples used by the standard and VBD methods.

1. Enable custom jitter buffer settings by setting the **arrisMtaDevCustomJitterBufferEnabled** object to **on(1)**. The default is **off(0)**.
When custom jitter buffer settings are enabled, the default settings are:
 - Minimum: 5 milliseconds
 - Nominal: 10 milliseconds
 - Maximum: 60 milliseconds
2. Configure the minimum, nominal, and maximum custom jitter buffer settings by setting the following objects. The valid range for all three objects is **5** to **135** (milliseconds).

arrisMtaDevCustomMinJitterBuffer

Minimum custom jitter buffer duration.

arrisMtaDevCustomNomJitterBuffer

Nominal custom jitter buffer duration.

arrisMtaDevCustomMaxJitterBuffer

Maximum custom jitter buffer duration.

Configuring Call Progress Tones

Use this procedure to configure call progress tones. Touchstone .EURO loads support provisioning of call progress tones through the **pkcSigDevToneTable**. North American loads support this table only when the country code setting (set by **ppCfMtaCountryTemplate**) is one of the North American country codes.

MIB Tables

Touchstone firmware uses two MIB tables to define call progress tones. See [Default Tone Settings](#) (page 271) for a list of default tone definitions for each country code type.

pkcSigDevToneTable

Defines the tone type, repeat count, and whether the last tone should be held steady after completing the cadence. Each entry in the table contains the following objects:

- **pkcSigDevToneType**: the index for the table (see below).
- **pkcSigDevToneWholeToneRepeatCount**: the number of times to repeat the entire sequence.
- **pkcSigDevToneSteady**: set to **true**(1) to keep the last tone in the sequence on until reaching the timeout.

pkcSigDevMultiFreqToneTable

Defines the actual frequencies for each tone defined in the tone table. Each entry in this table contains the following objects:

- **pkcSigDevToneNumber**: A secondary index, indicating the sequence number of the defined tone. Up to eight tones may be defined for a tone type.
- **pkcSigDevToneFirstFrequency**, **pkcSigDevToneSecondFrequency**, **pkcSigDevToneThirdFrequency**, **pkcSigDevToneFourthFrequency**: Up to four frequencies per defined tone. To disable a frequency, set it to **0**.
- **pkcSigDevToneFreqMode**: Determines how the frequencies define the tone:
 - **firstModulatedBySecond**(1): The first frequency is modulated by the second frequency, according to the percentage specified by **pkcSigDevToneFreqAmpModePrtg**. The third and fourth frequencies are ignored.
 - **summation**(2): All specified frequencies are added together without adding modulation.
- **pkcSigDevToneFreqAmpModePrtg**: The percentage of amplitude modulation to apply when using the **firstModulatedBySecond** setting.
- **pkcSigDevToneDbLevel**: The decibel level for each tone. The default is -40 dBm.
- **pkcSigDevToneOnDuration**: The time, in milliseconds, to play the defined tone.
- **pkcSigDevToneOffDuration**: The time, in milliseconds, of silence before the next tone.
- **pkcSigDevToneFreqRepeatCount**: The number of times to play the defined tone.

The **pkcSigDevToneType** object acts as the index for both tables. The index is one of the following values:

- **busy**(1)
- **confirmation**(2)
- **dial**(3)
- **messageWaiting**(4)
- **offHookWarning**(5)
- **ringBack**(6)
- **reOrder**(7)
- **stutterdial**(8)
- **callWaiting1**(9)
- **callWaiting2**(10)
- **callWaiting3**(11)
- **callWaiting4**(12)

The **pkcSigDevToneSteady** object, when set to **true**(1), keeps the last tone on.

The **pkcSigDevToneWholeToneRepeatCount** object defines how many times to repeat the on/off sequence.

The following tones are not supported in the MIB tables or through the supported line package:

- **alertingSignal**(13)
- **specialDial**(14)
- **specialInfo**(15)
- **release**(16)
- **congestion**(17)
- **userDefined1**(18)
- **userDefined2**(19)
- **userDefined3**(20)
- **userDefined4**(21)

Action

Follow these steps to configure call progress tones.

1. Modify the **pkcSigDevToneTable** table to define the repeat count and whether the last tone in the sequence is steady.
2. Modify the **pkcSigDevMultiFreqToneTable** to define the frequencies and duration of each tone in the sequence.
3. To modify a Call Waiting tone, follow the first two steps and then:
 - a. Set the **pkcNcsEndPntConfigCallWaitingDelay** object to define the amount of delay between repeats of the Call Waiting tones.



Note: The **pkcSigDevToneWholeToneRepeatCount** object is ignored for the Call Waiting tones.

- b. Set the **pktcNcsEndPntConfigCallWaitingMaxRep** object to define the repeat count for the Call Waiting tones.



Note: Do not use the **pktcSigDevMultiFreqToneTable** to configure the delay between repeated Call Waiting tones.

Gain Compensated Tone Generation

Touchstone firmware provides a patent-pending feature to automatically adjust FSK and CAS tone generation to compensate for MSO-selected loss value settings.

Touchstone firmware automatically adjusts FSK and tone levels to compensate for the loss plan values so that generated FSK and CAS tones are always within applicable specifications. You can make changes to the default levels as needed.

On-Hook vs. Off-Hook Gain

Touchstone firmware allows separate gain control for on-hook and off-hook levels. By default, the standard gain control MIB objects control levels for both conditions, but setting the **arrisMtaDevLevelControlOffHookEnable** object to **enable(1)** allows off-hook gain to be configured separately using the **arrisMtaDevLevelControlOffHookCAS** and **arrisMtaDevLevelControlOffHookFSK** objects.

arrisMtaDev...				
LevelControl			GainControl	
OffHookEnable	OffHookFSK	OffHookCAS	FSK	CAS
Disabled	Not used	Not used	Controls both on and off hook gains	Controls both on and off hook gains
Enabled	Controls off-hook gain	Controls off-hook gain	Controls only on-hook gain	Controls only on-hook gain

Action

Perform the following tasks as needed.

Configuring Gain Control using SNMP

Follow these steps as necessary to configure gain control settings.

1. Adjust eDVA-generated on-hook and default off-hook FSK tones (CID and VMWI) by setting the **arrisMtaDevGainControlFSK** object. Valid range: **−10** to **2** (dBm). Default: **0**.
2. Adjust the transmit digital gain adjustment for eDVA-generated on-hook and default off-hook CAS tone by setting the **arrisMtaDevGainControlCAS** object. Valid range: **−2** to **2** (dBm). Default: **0**.

3. To set off-hook FSK and CAS tone levels that are different from the on-hook levels:
 - a. Set the off-hook FSK tone level by setting the **arrisMtaDevLevelControlOffHookFSK** MIB object.
Valid range: **-32** to **-10** (dBm). Default: **-15**.
 - b. Set the off-hook CAS tone level by setting the **arrisMtaDevLevelControlOffHookCAS** MIB object.
Valid range: **-32** to **-10** (dBm). Default: **-15**.
 - c. Enable the off-hook gain settings by setting the **arrisMtaDevLevelControlOffHookEnable** MIB object to **enable(1)**. The default is **disable(0)**, which uses the established **arrisMtaDevGainControlFSK** and **arrisMtaDevGainControlCAS** objects for both on-hook and off-hook levels.
4. Adjust the transmit digital gain adjustment for eDVA-generated Call Progress tones (dial tone, busy tone, ringback, etc.) to the CPE by setting the **arrisMtaDevGainControlLocalTone** object. Valid range: **-2** to **2** (dBm). Default: **0**.
5. Adjust the transmit digital gain adjustment for eDVA-generated Call Progress tones (ringback) to the network by setting the **arrisMtaDevGainControlNetworkTone** object. Valid range: **-2** to **2** (dBm). Default: **0**.
6. Adjust the transmit digital gain adjustment for eDVA-generated DTMF tones to the CPE by setting the **arrisMtaDevGainControlLocalDTMF** object. Valid range: **-15** to **9** (dBm). Default: **0**.
7. Adjust the transmit digital gain adjustment for eDVA-generated DTMF tones to the network by setting the **arrisMtaDevGainControlNetworkDTMF** object. Valid range: **-9** to **9** (dBm). Default: **0**.

**CAUTION****Service affecting**

Changing the delta Rx/Tx Gain from the default value based on the country template used may affect overall voice transmission quality, local tone levels, digit detection, and modem/fax tone detection. PESQ scores may also be affected when additional loss is introduced.

8. Adjust the transmit digital gain adjustment for voice by setting the **arrisMtaDevGainControlTxVoice** object. Valid range: **-2** to **2** (dBm). Default: **0**.



Note: This setting does not affect local tone or FSK levels.

9. Adjust the receive digital gain adjustment for voice by setting the **arrisMtaDevGainControlRxVoice** object. Valid range: **-16** to **16** (dBm), or **-128** (the default) to use the eDVA-wide setting.



Note: Even though the deltaRx/Tx Gain MIB objects are defined to be line-based, setting the object for any valid line sets the delta gain for all lines. Also, if the same delta Rx/Tx

Gain object appears multiple times in the CM configuration file using different indexes, the eDVA uses the last instance to set the gain.

**CAUTION****Potentially service-affecting**

Setting endpoint gain too high or too low may disable the voice path. Lower settings beyond recommended levels may impact voice quality or fax or modem transmission.

10. Adjust the transmit digital gain adjustment for individual lines by setting the **arrisMtaDevEndPntGainControlTxVoice** object. Valid range: **-16** to **16** (dBm), or **-128** (the default) to use the eDVA-wide setting.
11. Adjust the receive digital gain adjustment for individual lines by setting the **arrisMtaDevEndPntGainControlRxVoice** object. Valid range: **-16** to **16** (dBm), or **-128** (the default) to use the eDVA-wide setting.

Provisioning Preset Downstream Frequencies

Use this procedure to provision one or more preset frequencies. This feature allows Touchstone E-UEs to quickly lock onto a known downstream during initial registration, reducing the initial time required for ranging and registering after installation. You can also clear the list of frequencies.

You provision preset downstream frequencies through SNMP using MIB objects.

Preset Frequency MIB Objects

The following MIB objects control preset frequencies. You can make changes to these objects using an SNMP manager or through the configuration file.

arrisCmDevPresetFrequency

Entries in a table of up to 20 preset frequencies.

arrisCmDevClearPresetFrequencies

Set to **true**(1) to clear the preset frequency table.

arrisCmDevClearCachedFrequencies

Set to **true**(1) to clear the cached frequencies.

Dial Pulse Support

Dial pulse support may be required to support subscriber equipment such as older rotary phones or alarm systems. Touchstone firmware provides two methods of dial pulse support:

- Direct relay—Touchstone eDVAs relay dial pulses to the CMS or P-CSCF.
- In-band tone relay (patent pending)—Touchstone E-UEs detect dial pulses and relay the information to the network as DTMF tones.
- Softswitch (CMS) dial pulse—Touchstone MTAs relay dial pulses to the CMS.

AR01.1 supports 20 pps dialing.

The support method used depends on the network configuration, and is selected by setting the ARRIS-proprietary [arrisMtaDevEndPntDialingMethod](#) MIB object as follows:

Value	Method	Description
1	Tone	(default) Enables DTMF detection only.
2	Pulse	Enables pulse dialing detection.
3	Tone & Pulse	Enables both DTMF and pulse dialing detection.
4	Pulse with DTMF Relay	Pulse dialing detection with DTMF in-band relay (gateway dial pulse, patent pending).
5	Tone & Pulse with DTMF Relay	Both DTMF and pulse dialing detection with DTMF in-band relay enabled (gateway dial pulse).

Note: Setting the [arrisMtaDevEndPntDialingMethod](#) object using an SNMP browser stores the value in non-volatile memory.

Inband DTMF Transmission

Touchstone firmware provides a feature to allow inband transmission of DTMF tones, even if a line is configured for pulse-only dialing. Once a call is established, Touchstone eDVAs pass all DTMF tones received from a CPE through the upstream voice path. eDVAs configured for pulse-only dialing operation drop any DTMF tones received before the call is fully established.

This feature allows a subscriber, provided with pulse-only dialing capabilities, to send DTMF tones to the far end. This might be used for calling card number entry, bank card data entry, automated service responses, and similar services.

Inband DTMF transmission is automatically enabled after a call has been established. No special configuration or changes to the dialing method are needed to enable this feature.



Note: If a line is provisioned as pulse-dialing only, the eDVA does not process received DTMF digits beyond treating them as audio to pass to the upstream voice path. If the CMS requests

that the eDVA notify the CMS of any collected digits, the received DTMF tones are not reported.

Action

Follow these steps to configure dial pulse support on Touchstone eDVAs.



Note: The **arrisMtaDevEndPntDialingMethod** object setting is stored in non-volatile memory when set through SNMP after the eDVA has completed registration.

1. To enable Gateway (IPDT) dial pulse support, set the **arrisMtaDevEndPntDialingMethod** object to **5** (pulse and DTMF detection).
2. To enable softswitch dial pulse support, set the **arrisMtaDevEndPntDialingMethod** object to **3** (pulse and DTMF detection).
3. To disable dial pulse detection, set the **arrisMtaDevEndPntDialingMethod** object to **1**. This may be necessary in certain situations where internal house wiring problems cause occasional “phantom” dial pulse digits.

Gateway Dial Pulse Example

The following configuration file provides an example of how to configure Gateway dial pulse support.

```

TelephonyConfigFileBeginEnd = 1
SnmpMib = pktcMtaDevEnabled.0 true
SnmpMib = arrisMtaDevEndPntDialingMethod.1 toneAndPulseWithDTMFRelay
SnmpMib = arrisMtaDevEndPntDialingMethod.2 toneAndPulseWithDTMFRelay
SnmpMib = pktcSigDefCallSigTos.0 0
SnmpMib = pktcSigDefMediaStreamTos.0 0
SnmpMib = pktcSigTosFormatSelector.0 ipv4TOSOctet
SnmpMib = pktcMtaDevRealmOrgName.DEV50 "Really Amazing Telephone Company"
SnmpMib = pktcNcsEndPntConfigCallAgentId.9 "ca@sn05.dev2"
SnmpMib = pktcNcsEndPntConfigCallAgentId.10 "ca@sn05.dev2"
SnmpMib = pktcNcsEndPntConfigCallAgentUdpPort.9 2727
SnmpMib = pktcNcsEndPntConfigCallAgentUdpPort.10 2727
SnmpMib = pktcMtaDevCmsKerbRealName.SN05.DEV2 "SWLAB. ATL. ARRIS"
SnmpMib = pktcMtaDevCmsIpsecCtrl.SN05.DEV2 true
SnmpMib = pktcNcsEndPntConfigMWD.9 10
SnmpMib = pktcNcsEndPntConfigMWD.10 10
SnmpMib = pktcMtaDevCmsUnsolicitedKeyNomTimeout.SN05.DEV2 20000
SnmpMib = pktcMtaDevRealmOrgName.SWLAB. ATL. ARRIS "Really Amazing Telephone Company"
TelephonyConfigFileBeginEnd = 255

```

Configuring Hook Flash Timing

Follow these steps to configure hook flash timing.



Note: These settings must be made in the eDVA configuration file.

Default Timing Settings

The country code template determines the default minimum and maximum flash timings, overriding the defaults specified by the PacketCable MIB. See [Country Code Templates](#) (page 268) for a list of default hook flash timings for each supported country code.

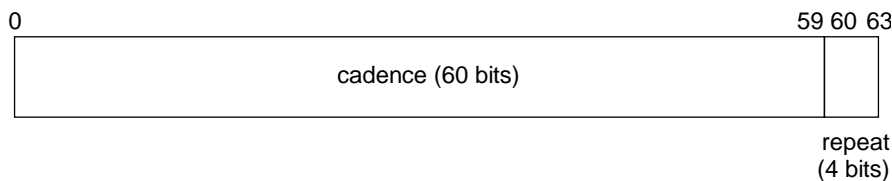
Action

Follow these steps to set hook flash timings.

1. In the eDVA configuration file, set the **pktcNcsEndPntConfigMinHookFlash** MIB object to the minimum time a line needs to be on hook for a valid hook flash. Use the **ifIndex** to specify the line; for example, **pktcNcsEndPntConfigMinHookFlash.9** specifies line 1. Valid range: **20** to **1550** milliseconds.
2. In the eDVA configuration file, set the **pktcNcsEndPntConfigMaxHookFlash** MIB object to the maximum time the line needs to be on hook for a valid hook flash. Valid range: **20** to **1550** milliseconds.

Provisioning Ring Cadences

Ring cadence and ring splash may be provisioned through the PKTC-IETF-MTA-MIB objects **pktcSigDevRgCadence** (standard ring), **pktcSigDevR0Cadence** through **pktcSigDevR7Cadence** (distinctive ringing), and **pktcSigDevRsCadence** (ring splash). These objects consist of the following fields:



The fields are:

cadence

60 bits representing the ring cadence. Each bit represents 100 ms; **1** represents tone and **0** represents silence. The pattern can be up to 6 seconds long; to specify a shorter pattern, begin the pattern with zeroes (for example, 15 zero bits at the beginning of the pattern would shorten the cadence to 4.5 seconds).

repeat

Specify either **0000** for a repeatable cadence, or **1000** for a non-repeatable cadence.

For example, a value of **00.00.0F.FF.FF.00.00.00** specifies a repeating ring cadence of 2 seconds on, 2 seconds off (the first two seconds of silence is discarded).



Note: The **pktcSigDevRsCadence** (ring splash) object must always be non-repeatable. The eDVA rejects any attempt to make the ring splash repeatable.

Post-Provisioning

Touchstone firmware provides the ability to post-provision lines and other device definitions, allowing you to add or remove service without restarting the eDVA.

NCS Post-Provisioning

Touchstone NCS loads are fully compliant with PacketCable eDVA Post-Provisioning functionality, described in section 7.6 of the *PacketCable MTA Device Provisioning Specification*, PKT-SP-PROV1.5-I04-090624. You can post-provision:

- CMS entries
- KDC realms
- Endpoints (lines)

Action

Follow these steps to post-provision a line on a Touchstone NCS load.

1. To add or remove a CMS, modify the **pktcMtaDevCmsTable** as required. You can add or remove entries in the table.
2. To add or remove a KDC realm, modify the **pktcMtaDevRealmTable** as required. You can add or remove entries in the table.
3. To add or remove an endpoint, modify the **pktcNcsEndPntConfigTable** as required. You can add or remove entries in the table.

Provisioning ARRIS SIP Loads

This chapter applies only to ARRIS SIP loads.

Overview of SIP Features

This section describes some important features of the SIP loads.

Barge-In

The **Join** header (RFC 3911) may be used to barge-in to an existing call. A brief tone is played to the existing call as the calls are conferenced together.

Loopback

The SIP eDVA can terminate loopback calls. When a loopback call is received, the phone does not ring. It automatically answers the call and loops back media or packets to the originator. The eDVA has a 2 call per line resource limitation. The user may make a call or receive a call while a loopback call is in progress. If another call is placed, either by an incoming call or a new outbound call, the loopback call is disconnected.

Packet Loopback is analogous to NETWLOOP, and controlled by the **sipCfgPacketLoopbackNumber** object. Media Loopback is analogous to NETWTEST, and controlled by the **sipCfgMediaLoopbackNumber** object.

The *Loopback Draft* (<http://www.ietf.org/internet-drafts/draft-ietf-mmusic-media-loopback-08.txt>) specifies the CODEC negotiation involved in setting up a loopback call.

Touchstone firmware also supports a proprietary loopback method. If the eDVA receives a call from a number matching those provisioned in specific MIB objects, it creates a loopback call.

Extended Offhook Processing

Touchstone firmware partially implements extended offhook processing as defined in PKT-SP-RSTF-I08-110127, section 7.1.4.4. Touchstone firmware implements extended offhook processing as shown below.

Origination Mode

Origination mode typically involves the subscriber picking up the phone, then either not dialing a number or pausing too long between digits. PacketCable defines an Origination Mode Dial Time timer and a Long Interdigit Timer (typically 16 seconds each); either timer expiring invokes the “permanent sequence.”

Use the `pktcEUERSTNfBCallOrigDTTimer` object to set the Dial Time timer. The default time is 16 seconds.

Use the `pktcEUERSTNfBCallOrigModLongIntDig` object to set the Long Interdigit timer. The default time is 16 seconds.

Termination Mode

When the eDVA receives the BYE signal, it starts a timer. Prior to TS7.6 MSUP2, the timer is fixed at 20 seconds. In TS7.6 MSUP2 and newer loads, the `sipCfgTermOffHookProcessingDelay` object controls the timer. The valid range for the timer is 0 to 60 seconds; the default is 20 seconds.

If the timer expires before the subscriber hangs up, the eDVA invokes the “permanent sequence.”

Permanent Sequence

A series of MIB objects, defined in the CL-PKTC-EUE-RST-MIB, specify the permanent sequence played in response to an extended offhook. The default behavior is:

D11PLUS loads:

1. Reorder tone for 30 seconds
2. OSI for 1 second
3. Silence for 4 seconds
4. Howler tone for 60 seconds

Other loads:

1. OSI for 1 second
2. Silence for 10 seconds
3. Howler tone for 60 seconds

If the subscriber has not hung up the phone by the time the sequence ends, the line enters a lockout state until the subscriber hangs up.

The following table shows the MIB objects used to provision the permanent sequence and the default values for each object.

Object	.TW Loads	Other Loads
pktcEUERSTNfBCallPermSeqTone1	"file:///PacketCableRST/ro"	"file:///PacketCableRST/osi"
pktcEUERSTNfBCallPermSeqTimer1	30	1

Object	.TW Loads	Other Loads
pkcEUESTNfBCallPermSeqTone2	"file:///PacketCableRST/osi"	"file:///PacketCableRST/nt"
pkcEUESTNfBCallPermSeqTimer2	1	10
pkcEUESTNfBCallPermSeqTone3	"file:///PacketCableRST/nt"	"file:///PacketCableRST/ot"
pkcEUESTNfBCallPermSeqTimer3	4	60
pkcEUESTNfBCallPermSeqTone4	"file:///PacketCableRST/ot"	""
pkcEUESTNfBCallPermSeqTimer4	60	0

Emergency Calls

Any number matching the pattern associated with the digit map action **EMERGENCY- CALL** is treated as an emergency call. Emergency calls have special treatment. The subscriber is, by default, not allowed to terminate an emergency call (this can be overridden by setting the **sipCfgSipFeatureSwitch** bit **0x00000100**). If the user goes on hook, and the default behavior is in effect, the eDVA does not send a BYE message. Instead, the call is put on “Network Hold” as follows:

1. When the subscriber goes on-hook during an emergency call, the Touchstone eDVA sends an INVITE (**pri ori ty: emergency SDP: a=i nact i ve**). This causes the PSAP operator to hear a tone that indicates that the user went on-hook.
2. The PSAP operator can send an INVITE (**pri ori ty: emergency SDP: a=sendrecv**) that causes the eDVA to ring. If the subscriber goes off-hook before receiving such an INVITE, then it should send the invite to re-establish two-way communications.
3. The Network Hold Timer specifies the maximum time that an emergency call is preserved in the Network Hold state. The timer is started every time that the user goes on-hook during an emergency call, and is cleared if the user goes off hook. If the Network Hold Timer expires, then the eDVA sends a BYE message to finally terminate the call. The default value for the Network Hold Timer is 45 minutes.

Emergency calls have the following restrictions:

- Outgoing emergency calls offer only the G.711 CODEC in the SDP.
- Caller ID blocking is overridden on outgoing emergency calls.
- In the INVITE, the eDVA does not include the route received in the **Servi ce- Route:** header that was received in the registration response.
- Incoming emergency calls cannot be transferred.

Incoming calls are treated as emergency calls under either of the following conditions:

- The **P- Asserted- Identi ty:** header value matches the value of the **sipCfgEmergencyServiceURN** object (the default is **URN: servi ce: sos**). If you change this value, always begin the string with the **URN:** prefix to comply with RFC 5031.
- The **Priority** header value is “emergency.”

To disconnect any active call when originating or receiving an emergency call, set bit **0x00001000** in the **sipCfgSipFeatureSwitch**.

Distinctive Ringing

Several distinctive ringing types, corresponding to R0 through R7, are defined by the country template. If the `Alert-Info:` header is received, Touchstone firmware compares the header to the strings specified in the MIB objects [sipCfgAlertInfoR0](#) through [sipCfgAlertInfoR7](#). If the header matches one of these, the eDVA plays the corresponding R0–R7 tone. AR01.1 also supports tones WT1 through WT4 for use with Call Waiting, and maps R1 through R4 to WT1 through WT4 for a Call Waiting alert.

The following configuration shows the default string settings.

```
{ SnmpMib sipCfgAlertInfoR0.0 "< http://127.0.0.1/Bellcore-dr0 >" }
{ SnmpMib sipCfgAlertInfoR1.0 "< http://127.0.0.1/Bellcore-dr1 >" }
{ SnmpMib sipCfgAlertInfoR2.0 "< http://127.0.0.1/Bellcore-dr2 >" }
{ SnmpMib sipCfgAlertInfoR3.0 "< http://127.0.0.1/Bellcore-dr3 >" }
{ SnmpMib sipCfgAlertInfoR4.0 "< http://127.0.0.1/Bellcore-dr4 >" }
{ SnmpMib sipCfgAlertInfoR5.0 "< http://127.0.0.1/Bellcore-dr5 >" }
{ SnmpMib sipCfgAlertInfoR6.0 "< http://127.0.0.1/Bellcore-dr6 >" }
{ SnmpMib sipCfgAlertInfoR7.0 "< http://127.0.0.1/Bellcore-dr7 >" }
```

Touchstone firmware supports both the Bellcore-defined tones shown in the listing above, and the equivalent PacketCable 2.0-defined tones file: `///PacketCableRST/rn`, where *n* is 0 through 7.

SIP Provisioning Considerations

This section describes SIP provisioning considerations.

Information Required for SIP

The SIP load requires three pieces of information to function properly:

Outbound P-CSCF

The destination device for all outbound messages. The setting is used as the domain in the Request-URI for all outgoing INVITE messages. The E-UE supports a “global” proxy that applies to all lines, and a per-line proxy that applies to a single line. See [“Configuring Per-Line Proxy and Registrar”](#) (page 98) for details.

Registrar

Registration messages are sent to the outbound proxy’s IP address, but the Request-URI address is provisioned as the registrar address. This setting may or may not be the same as the outbound proxy setting. The E-UE supports a “global” registrar that applies to all lines, and a per-line registrar that applies to a single line. See [“Configuring Per-Line Proxy and Registrar”](#) (page 98) for details.

Domain Settings

The domain is set as part of the DHCP process of the Telephony Modem using DHCP Option 15. This domain is used in all **to** and **from** URIs that the Telephony Modem generates.

The following are examples of REGISTER and INVITE messages with the following provisioning when bit **0x04000000** of the SIP Feature Switch is not set (bit value = 0):

```

Domain          arri s-i. org
sipCfgProxyAdr   ser. arri s-i. org; 5060
sipCfgRegistrarAdr registrar. arri s-i. org; 5060
User-id (phone #) 7705552001
Called Number     7705552002
REGISTER sip: registrar. arri s-i. org; 5060 SIP/2. 0
From: "SIP1 Line1"< sip: 7705552001@arri s-i. org>; tag=94b73228- a013d16- 13c4-
20- 21829ddd- 20
To: "SIP1 Line1"< sip: 7705552001@arri s-i. org>
Call-ID: 94b6e3b0- a013d16- 13c4- 20- 22ce4f5- 20
CSeq: 1 REGISTER
Via: SIP/2. 0/UDP 10. 1. 61. 22: 5060; branch=z9hG4bK- 20- 7ed6- 6787cc6b
Allow: INVITE, ACK, BYE, CANCEL, NOTIFY
Max-Forwards: 70
Contact: "SIP1 Line1"< sip: 7705552001@10. 1. 61. 22: 5060>
Content-Length: 0
INVITE sip: 7705552002@ser. arri s-i. org; 5060 SIP/2. 0
From: "SIP1 Line1"< sip: 7705552001@arri s-i. org>;
tag=94b73808- a013d16- 13c4- 62- 4e4a1883- 62
To: < sip: 7705552002@arri s-i. org>
Call-ID: 94b6f7d8- a013d16- 13c4- 62- 5370cb92- 62@arri s-i. org
CSeq: 1 INVITE
Via: SIP/2. 0/UDP 10. 1. 61. 22: 5060; branch=z9hG4bK- 62- 180b5- 631cc050
Allow: INVITE, ACK, BYE, CANCEL, NOTIFY
Max-Forwards: 70
Contact: < sip: 7705552001@10. 1. 61. 22: 5060>
Content-Type: application/SDP
Content-Length: 168
<SDP REMOVED>

```

Here are examples of REGISTER and INVITE with IP address-based provisioning with bit **0x04000000** set (bit value = 1):

```

Domain arri s-i. org
sipCfgProxyAdr 10. 1. 63. 10; 5060
sipCfgRegistrarAdr 10. 1. 63. 11; 5060
sipCfgSipFeatureSwitch 0x04000000
User-id (phone #) 7705552001
Called Number 7705552002
REGISTER sip: 10. 1. 63. 11: 5060 SIP/2. 0
From: "SIP1 Line1"< sip: 7705552001@10. 1. 63. 10: 5060>; tag=94b73228- a013d16-
13c4- 20- 21829ddd- 20
To: "SIP1 Line1"< sip: 7705552001@10. 1. 63. 10: 5060>
Call-ID: 94b6e3b0- a013d16- 13c4- 20- 22ce4f5- 20
CSeq: 1 REGISTER
Via: SIP/2. 0/UDP 10. 1. 61. 22: 5060; branch=z9hG4bK- 20- 7ed6- 6787cc6b
Allow: INVITE, ACK, BYE, CANCEL, NOTIFY
Max-Forwards: 70
Contact: "SIP1 Line1"< sip: 7705552001@10. 1. 61. 22: 5060>
Content-Length: 0
INVITE sip: 7705552002@10. 1. 63. 10: 5060 SIP/2. 0
From: "SIP1 Line1"< sip: 7705552001@10. 1. 63. 10: 5060>; tag=94b73808- a013d16-
13c4- 62- 4e4a1883- 62
To: < sip: 7705552002@10. 1. 63. 10: 5060>
Call-ID: 94b6f7d8- a013d16- 13c4- 62- 5370cb92- 62@10. 1. 63. 10
CSeq: 1 INVITE
Via: SIP/2. 0/UDP 10. 1. 61. 22: 5060; branch=z9hG4bK- 62- 180b5- 631cc050
Allow: INVITE, ACK, BYE, CANCEL, NOTIFY
Max-Forwards: 70
Contact: < sip: 7705552001@10. 1. 61. 22: 5060>

```

Content-Type: application/SDP Content-Length: 168
<SDP REMOVED>

SIP Registration Behavior

All SIP Touchstone firmware loads use the same algorithm for normal registration and re-registration:

- Registration begins after a randomized wait time between 1 and (MaxWaitDelay) seconds. The default for MaxWaitDelay is 10 minutes; the **pktnCsEndPntConfigMWD** object controls the delay time (in seconds). Note that line 1 uses index 9, line 2 uses index 10, and so on.
- By default, the registrar sends the registration in the 200 OK response to a REGISTER request. The **sipCfgRegExpires** object can specify a suggestion to the registrar for a desired registration expiry value. If the specified value is non-zero, the eDVA uses the smaller of the expiry value returned by the registrar in the 200 OK or the value specified in **sipCfgRegExpires**. If **sipCfgRegExpires** is unconfigured or specified to zero, no expires value is specified in the REGISTER request.
- After registration, the eDVA attempts to re-register after a random time between 50% and 75% of the expiration time.
- If registration is unsuccessful (for example, the eDVA receives no response or a 401 response), the eDVA retries registration using a backoff algorithm at intervals specified by RFC 3261.

The backoff algorithm is exponential. The initial backoff starts at the value specified by **sipCfgRegTimerMin** (default: 60 seconds). The interval between retry attempts doubles until it is greater than the value specified by **sipCfgRegTimerMax** (default: 1800 seconds). The interval between retries is not to exceed the value specified by **sipCfgRegTimerMax**. The eDVA uses the following formula to calculate the delay:

$$\text{time} = \min(\text{TimerMax}, (\text{TimerMin} \times 2^{(\#\text{failures}-1)}))$$

The default values are 60 and 1800 seconds, and can be changed by specifying new values in the eDVA configuration file.

Each REGISTER attempt is not a single message. The eDVA will retry REGISTER messages based on the retransmission algorithm specified in RFC 3261. By default, the eDVA retries 7 times over a 32-second interval. The retransmission algorithm is also exponential. The basis interval is specified by T1 (**sipCfgT1**). The total time to retransmit is specified by TimerF (**sipCfgTimerF**).

SIP Feature Switch

The SIP feature switch enables or disables extended features in the SIP load. See “SIP Feature Switch” for a detailed description of each switch.

Provisioning Details

This section describes various SIP features that can be configured in the provisioning file.

Digit Map

SIP digit maps are identical to NCS digit maps:

- An **x** indicates any digit.
- The **|** (pipe) character separates different entries.

After each digit is pressed, the digit collector looks at each entry in the digit map to determine if any entry is complete. If an entry is complete, dialing is complete. There are some exceptions when star codes are specified. For a star code (VSC) to be handled, it must be in the digit map and possibly in another MIB object.

The below example is 10 digit dialing with 2 digit star codes.

```
{Snmplib sipCfgDigitMap. 0 "xxxxxxxxxx|*xx"}
```

Proxy Address

The proxy address specifies the destination where the eDVA sends all SIP requests. You must also specify the proxy type, IP or DNS.

```
{Snmplib sipCfgProxyAdr. 0 "ser.arri s-i . org; 5060"}  
{Snmplib sipCfgProxyType. 0 dns}
```

Registrar Address

The registrar address specifies the text of the top line of the REGISTER request.

```
{Snmplib sipCfgRegistrarAdr. 0 "registrar.arri s-i . org; 5060"}  
{Snmplib sipCfgRegistrarType. 0 dns}
```

Packetization Rate

The packetization rate can be set to 10 or 20 milliseconds.

```
{Snmplib sipCfgPacketizationRate. 0 20}
```

Provisioned CODEC Array

A semicolon-delimited list of CODEC types. This only affects outbound calls. The eDVA attempts to negotiate the CODECs in the list. CODECs are listed in order of preference and used only on initial outbound INVITE messages. Incoming requests answer with the CODECs supported in the initial offer's order.

```
{Snmplib sipCfgProvisionedCodecArray. 0 "PCMU; PCMA; G729; telephone-event"}
```

Repeat Dialing

The following MIB objects control repeat dialing.

sipCfgRepeatDialingInterval

Specifies the time between attempts to connect to other party.

sipCfgRepeatDialingTimeout

A timer value that specifies how long the eDVA should keep trying to connect to the other party.

sipCfgRepeatDialingSessionProgressTimer

Specifies the length of time after a 183 is received before considering the repeat dialing attempt successful. Often a 183 is received prior to a negative response. This timer prevents false positives.

These are the default values.

```
{ SnmpMib sipCfgRepeatDialingInterval.0 30}
{ SnmpMib sipCfgRepeatDialingTimeout.0 1800}
{ SnmpMib sipCfgRepeatDialingSessionProgressTimer.0 2}
```

Call Forwarding Forbidden Numbers

There are certain phone numbers (eg. 911) that Call Forwarding should not accept as forwarding numbers. The **sipCfgCallForwardForbiddenNumbers** MIB object specifies which numbers are not allowed as forwarding numbers. This object can be set only in the configuration file. Multiple numbers may be specified separated by | as follows:

```
{ SnmpMib sipCfgCallForwardForbiddenNumbers.0 "911|900|1900|0"}
```



Note: The emergency number set in **sipCfgEmergencyNumber** is also not allowed as a forwarding number.

Call Waiting setting Persistent Across Reboot

If the **sipCfgCallWaitingStarCodeSurvivesReset** MIB object is set to **true**, the call waiting state is set in non-volatile memory to ensure the state persists if the eDVA reboots.

```
{ SnmpMib sipCfgCallWaitingStarCodeSurvivesReset.0 true}
```

To clear this setting, to re-locate the MTA to another customer that may have a different preference or for some other reason, use the **sipCfgResetCallWaitingStarCode** MIB object to clear the NVM setting. The MIB object controls the setting of this value on a per-line basis, using a bit string to indicate which lines should be reset.

```
# reset lines 1 and 2
{ SnmpMib sipCfgResetCallWaitingStarCode.0 0x00000003}
```

Default G.711

When using G.729 as the primary voice CODEC, the eDVA must switch to a G.711 CODEC to support faxes transmission. Touchstone firmware switches to PCMU in this situation by default. Some customers require PCMA as the pass-through CODEC.

Use the [sipCfgDefaultG711](#) MIB object to specify a different pass-through CODEC, as follows:

```
{SnmpMib sipCfgDefaultG711.0 pcma}
```

Domain Override

The [sipCfgDomainOverride](#) MIB object specifies the string to use in the domain in all outbound SIP signaling messages.

```
{SnmpMib sipCfgDomainOverride.0 "arris-i.org"}
```

Emergency Calls

Emergency calls have special treatment. This special treatment is determined based on the outbound dialed string. Provision the emergency number using the [sipCfgEmergencyNumber](#) object, as follows:

```
{SnmpMib sipCfgEmergencyNumber.0 911}
```

The end user is never allowed to terminate an emergency call. If the user goes on hook, the eDVA does not send a BYE message. Instead, the call is put on "Network Hold" as follows:

1. When the end user goes on-hook during an emergency call, the Touchstone eDVA sends an INVITE (`priority: emergency SDP: a=inactive`). This causes the PSAP operator to hear a tone that indicates that the user went on-hook.
2. The PSAP operator can send an INVITE (`priority: emergency SDP: a=sendrecv`) that causes the eDVA to ring. If the user goes offhook before receiving such an INVITE, then it should send the invite to re-establish two-way communications.
3. The Network Hold Timer specifies the maximum time that an emergency call is preserved in the Network Hold state. The timer is started every time that the user goes on-hook during an emergency call, and is cleared if the user goes off hook. If the Network Hold Timer expires, then the eDVA sends a BYE message to finally terminate the call. The default value for the Network Hold Timer is 45 minutes.

See PKT-SP-RSTF-I08-110127 section 8.5.5.8 for more details on Network Hold for Emergency Calls, including call flows for various Network Hold scenarios.

Pulse Dialing

The [arrisMtaDevEndPntDialingMethod](#) MIB object enables pulse dialing. The object is indexed by line number.

```
{SnmpMib arrisMtaDevEndPntDialingMethod.1 toneAndPulse}
```

Line Specific Features

The [sipCfgPortFeatureSettings](#) MIB object enables various line-specific features. The object is a bit-mask; each bit controls a specific feature as shown in the following table.

Bit	Feature
0x80	Default
0x40	Caller ID disabled
0x20	Anonymous call rejection
0x10	Call Waiting disabled
0x08	Disable Three-Way Calling
0x04	Disable Caller ID display
0x02	Enable Call Transfer

Example:

```
{SnmpMib sipCfgPortFeatureSettings.1 02}
```

Distinctive Ringing

Several distinctive ringing types, corresponding to R0 through R7, are defined by the country template. If the Alert-Info: header is received, Touchstone firmware compares the header to the strings specified in the MIB objects [sipCfgAlertInfoR0](#) through [sipCfgAlertInfoR7](#). If the header matches one of these, the eDVA plays the corresponding R0–R7 tone.

The following configuration shows the default string settings.

```
{SnmpMib sipCfgAlertInfoR0.0 "< http://127.0.0.1/Bellcore-dr0 >"}
{SnmpMib sipCfgAlertInfoR1.0 "< http://127.0.0.1/Bellcore-dr1 >"}
{SnmpMib sipCfgAlertInfoR2.0 "< http://127.0.0.1/Bellcore-dr2 >"}
{SnmpMib sipCfgAlertInfoR3.0 "< http://127.0.0.1/Bellcore-dr3 >"}
{SnmpMib sipCfgAlertInfoR4.0 "< http://127.0.0.1/Bellcore-dr4 >"}
{SnmpMib sipCfgAlertInfoR5.0 "< http://127.0.0.1/Bellcore-dr5 >"}
{SnmpMib sipCfgAlertInfoR6.0 "< http://127.0.0.1/Bellcore-dr6 >"}
{SnmpMib sipCfgAlertInfoR7.0 "< http://127.0.0.1/Bellcore-dr7 >"}

```

Dialing Features

The [sipCfgDialFeatTable](#) specifies dialing features that are handled by the eDVA. The following example assigns Anonymous Call Reject to the ***40** dialing code.

```
# Set up Anonymous Call Rejection as star code *40
{SnmpMib sipCfgDialFeatName.1 anonCallReject}
{SnmpMib sipCfgDialFeatCode.1 "*40" }
{SnmpMib sipCfgDialFeatTone.1 stutterTone}
# Works on first 4 lines
{SnmpMib sipCfgDialFeatActive.1 0.0.0.F}
# optional -- currently only used for
# repeat dialing (where its value should be 02)
{SnmpMib sipCfgDialFeatMode.1 01}

```

See “Supported Dialing Features” for a list of eDVA-supported dialing features.

Hybrid features, which are passed up to the proxy in an INVITE message, are controlled by the [sipCfgDialProxyTable](#). The following example assigns ***50** as a hybrid feature.

```
# Active star code *50 as a hybrid feature
{SnmpMib sipCfgDialProxyCode.1 "*50" }

```



```
{Snmplib sipCfgDialProxyTone.1 stutterTone}
# Works on first 4 lines
{Snmplib sipCfgDialProxyActive.1 0.0.0.F}
# optional -- default is to use INVITE,
# may be used to send REFER instead.
{Snmplib sipCfgDialProxyMessageType.1 01}
```

Proxy features are sent directly to the proxy in an INVITE message. No configuration is necessary to handle these features. The proxy itself defines what feature is associated with a particular dialing code.

Call Transfer

Uses the target dialog (RFC 4538) if supported by far-end; otherwise, sends REFER on the existing Call Leg.

In most cases, call transfer is initiated by the pivot phone going on hook. This triggers a transfer if Call Transfer is enabled, and the call is in one of the **threeWayCalling**, **callingHolding**, or conference states. When advanced flash digit handling is enabled, call transfer can be initiated by the **4** digit.

The transfer is performed by the pivot phone. The pivot phone sends a REFER message to the original call. This REFER message includes a ReferTo: header which notifies the party being transferred who to contact. Embedded in the ReferTo: header is a Replaces: header (RFC 3891). The party receiving the REFER message then sends an INVITE to the party referenced in the ReferTo: header. The Replaces: header is then copied to its own header in this INVITE. The party receiving the INVITE with the Replaces: header uses this header to determine which call to disconnect and replace with the transferred party.

Feature Capabilities

Ten MIB objects allow control of certain features on a per-line basis. The objects follow a naming convention of **sipCfgfeatCapability**, where *feat* is the specific feature. Each MIB object consists of a hexadecimal bit string (32 bits wide) where each bit is a flag corresponding to a different line. These objects can be only set in the configuration file.

When a feature capability of a particular line is set to zero (off), that feature is disabled for that line regardless of the value of any other MIB object, including **sipCfgFeatureSettings**. By default, all feature capabilities are enabled for every line; all ten MIB objects default to the hex value **FFFFFFFF** (which is a string of 32 ones). When a feature is enabled, the user may still enable or disable the feature using dial codes. However, if the feature capability is disabled in the configuration file, the dial code to enable that feature does not work.

The following table lists the controllable features and the corresponding MIB object controlling its capability:

Feature	MIB Object
Caller ID Display	sipCfgCallerIdDisplayCapability
Caller ID Send	sipCfgCallerIdSendCapability

Feature	MIB Object
Anonymous Call Rejection	sipCfgAnonCallRejectionCapability
Call Waiting	sipCfgCallWaitingCapability
Three Way Calling	sipCfgThreeWayCallCapability
Call Transfer	sipCfgCallTransferCapability
Call Forwarding	sipCfgCallForwardCapability
Call Return	sipCfgCallReturnCapability
Call Redial	sipCfgCallRedialCapability
Call Holding	sipCfgCallHoldCapability

Examples:

To disable three-way calling for lines 1 and 3, add the following line to the configuration file:

```
{Snmplib sipCfgThreeWayCallCapability.0 FFFFFFFFA}
```



Note: For an 8- or 12-line Telephony Modem, this is identical to setting the value to 00000FFA because only the 12 least significant bits are used.

To disable three-way calling for all lines on a Telephony Modem, modify the setting as follows:

```
{Snmplib sipCfgThreeWayCallCapability.0 00000000}
```

To enable three-way calls for all lines, delete any settings for [sipCfgThreeWayCallCapability](#) from the configuration file and reset the modem.

For any of these changes to take effect, the modem needs to be reset to download the updated configuration file.

Timers

- Timer T1 — [sipCfgT1](#) (PacketCable 2.0 MIB also specified); specifies the initial interval between retransmission attempts.
- Timer T2 — [sipCfgT2](#) (TS7.6 MSUP2 and newer, PacketCable 2.0 MIB also specified); specifies the maximum retransmit interval for non-INVITE requests and INVITE responses. See RFC 3261 for details.
- Timer T4 — PacketCable 2.0 MIB; see RFC 3261 for details.
- Timer B — [sipCfgTimerB](#) specifies the number of retransmission attempts or the time to keep retransmitting INVITE messages. The [sipCfgMaxRetrans](#) object (deprecated) specifies the number of retries.
- Timer F — [sipCfgTimerF](#) specifies the time to keep retransmitting non-INVITE messages.
- Timers D and H — [sipCfgInviteLinger](#)
- Timer K — [sipCfgGenLinger](#)

Minimal Example

The following configuration file fragment provides a minimal example of SIP configuration.

```
set mcns_config_params {
{TelephonyConfigFileBeginEnd 1}
# eDVA Enabled
{Snmplib pktcMtaDevEnabled.0 1}
# Proxy Address
{Snmplib sipCfgProxyAdr.0 "ser.arri-s-i.org; 5060"}
{Snmplib sipCfgProxyType.0 1}
# Registrar Address -- used in REGISTER request URI
{Snmplib sipCfgRegistrarAdr.0 "registrar.arri-s-i.org; 5060"}
{Snmplib sipCfgRegistrarType.0 1}
#### Line 1 Configuration
# Phone Number
{Snmplib sipCfgPortUserName.1 7705558001}
# Caller-ID Display
{Snmplib sipCfgPortDisplayName.1 "David Line1"}
# Proxy Authentication Username
{Snmplib sipCfgPortLogin.1 traff}
# Proxy Authentication Password
{Snmplib sipCfgPortPassword.1 password}
# Line 1 Enabled
{Snmplib ifAdminStatus.9 1}
# Try registration time within 10 seconds of getting config file
{Snmplib pktcNcsEndPntConfigMWD.9 10}
#### Line 2 Configuration
# Phone Number
{Snmplib sipCfgPortUserName.2 7705558002}
# Caller-ID Display
{Snmplib sipCfgPortDisplayName.2 "David Line2"}
# Proxy Authentication Username
{Snmplib sipCfgPortLogin.2 traff}
# Proxy Authentication Password
{Snmplib sipCfgPortPassword.2 password}
# Line 2 Enabled
{Snmplib ifAdminStatus.10 1}
# Try registration time within 10 seconds of getting config file
{Snmplib pktcNcsEndPntConfigMWD.10 10}
{TelephonyConfigFileBeginEnd 255}
}
```

Provisioning SIP Support

AR01.1 provides SIP support for eDVAs. This feature requires modifications to the CM configuration file and a new eDVA configuration file. This feature requires a specialized firmware load.

Per-Line Proxy/Registrar Objects

The following table describes MIB objects for per-line proxy/registrar support. These objects must be set in the configuration file to be effective. For details about the SIP registration process, see [“SIP Registration Behavior”](#) (page 84)."

MIB Object		Description (Per-line)
Global	Per-line	
sipCfgRegistrarAdr	sipCfgPortRegistrarAdr	Registrar Server
	sipCfgPortRegistrarPort	Registrar Server port. The per-line setting is only valid if paired with a setting for sipCfgPortRegistrarAdr .
sipCfgRegistrarType	sipCfgPortRegistrarType	The type of address specified by the sipCfgPortRegistrarAdr object. It is only valid if paired with a setting for sipCfgPortRegistrarAdr . When both objects are defined in the configuration file, this value overrides the setting in the sipCfgRegistrarType for this line. Valid values for this setting are ipv4 (0) and dns (1).
sipCfgProxyAdr	sipCfgPortProxyAdr	Proxy Server address
	sipCfgPortProxyPort	Proxy Server port. It is only valid if paired with a setting for sipCfgPortProxyAdr .
sipCfgProxyType	sipCfgPortProxyType	The type of address specified by the sipCfgPortProxyAdr object. It is only valid if paired with a setting for sipCfgPortProxyAdr . When both objects are defined in the configuration file, this value overrides the setting in sipCfgProxyType for this line. Valid values for this setting are ipv4 (0) and dns (1).

Use the line number as the index for each object; for example, **sipCfgPortProxyAdr.2** specifies the proxy IP address for line 2.



Note: Per-line proxy/registrar is a *device*-only change; it cannot be changed using post-provisioning.

T.38 Provisioning Overview

The MIB object **sipCfgPortT38Mode** controls T.38 behavior. It allows the following values:

Value	Description
t380ff (1)	(default) Disables T.38.
t38Loose (2)	Enables T.38 Loose mode. In loose mode, the eDVA can use T.38 for fax transmission whether or not the remote endpoint has indicated T.38 support.
t38Strict (3)	Enables T.38 Strict mode. In strict mode, the eDVA can use T.38 for fax transmission only if the far end indicated support for T.38 during session negotiation.

The MIB object **arrisMtaDevEndPntFaxOnlyLineTimeout** configures fax-only mode for a line. This value of this object specifies the time, in seconds, to wait for fax or modem tones after receiving the SDP from the remote endpoint. If the time expires before detecting fax or modem tones, the eDVA drops the call. Valid range: **0** to **600**.

The MIB object **sipCfgPortMaxT38HSRedLevel** sets the maximum high-speed redundancy level used for T.38 fax relay, in both send and receive directions. The value of this object is the number of older data packets included in each T.38 datagram when transferring fax data. The actual redundancy level used is determined by negotiation with the remote endpoint. Valid range: **0** to **2**. Default: **1**.

Global Call Feature Control

The **sipCfgPortFeatureSettings** object allows you to control operation of the following call features:

- outbound Caller ID
- anonymous call rejection
- call waiting
- three-way calling

The **sipCfgPortFeatureSettings** object is structured as a collection of bit flags, as shown in the following table. The default value is **0**.

Bit	Description
0x40	callerIdPermanentDisable Set this bit to set the default outbound Caller ID method to "restrictive." The default setting presents Caller ID.
0x20	anonCallRejectionEnable Set this bit to enable anonymous call rejection. The subscriber can use a "star" code to disable anonymous call rejection if desired. The default setting permits anonymous calls.
0x10	callWaitingPermanentDisable Set this bit to disable Call Waiting. The subscriber can use a "star" code to enable Call Waiting if desired. The default setting is to enable Call Waiting.
0x08	threeWayCallingDisable Set this bit to disable hook flash processing during an active call. The default setting is to allow hook flash processing.
0x04	callIdReceiptDisable Set this bit to disable local CallerID display.
0x02	callTransferEnable Set this bit to enable Call Transfer. The bit can only be set in the configuration file.

Two objects control the persistence of Call Waiting settings:

sipCfgCallWaitingStarCodeSurvivesReset

Set this object to **true**(2) to enable storage of the Call Waiting Permanent Disable state in non-volatile memory. The default is **false**(1).



Note: If a subscriber disables Call Waiting (using a star code) with this object enabled, and the Telephony Modem is subsequently reissued to another subscriber, the new subscriber may assume that Call Waiting is disabled.

sipCfgResetCallWaitingStarCode

Set this object to **0xFFFFFFFF**, using an SNMP browser, to clear the Call Waiting Permanent Disable state from non-volatile memory for all lines.

Per-line Call Feature Control

Ten MIB objects provide per-line control over common calling features, allowing subscribers to order each feature separately. Each object is a map of 32 bits; the least significant bit represents line 1. Setting a bit to **0** disables the corresponding feature for that line.

These MIB objects must be set in the eDVA configuration file, and are not persistent across reboots. The default value for all these objects is **0xFFFFFFFF** (feature enabled on all lines).

Feature	MIB Object
Caller ID Display	sipCfgCallerIdDisplayCapability
Caller ID Send	sipCfgCallerIdSendCapability
Anonymous Call Rejection	sipCfgAnonCallRejectionCapability
Call Waiting	sipCfgCallWaitingCapability
Three-way Calling	sipCfgThreeWayCallCapability
Call Transfer	sipCfgCallTransferCapability
Call Forwarding	sipCfgCallForwardCapability
Call Return	sipCfgCallReturnCapability
Call Redial	sipCfgCallRedialCapability
Call Hold	sipCfgCallHoldCapability

Each Telephony Modem ignores bits beyond its line capacity. For example, on a TM702 Telephony Modem, **0x00000003** is equivalent to **0xFFFFFFFF**.

Action

Perform the following tasks as necessary:

- [CM Configuration File Changes](#) 95
- [eDVA Configuration File Changes](#) 95
- [Setting up Timers](#) 97
- [Configuring Per-Line Proxy and Registrar](#) 98
- [Specifying a SIP Domain Name](#) 99

CM Configuration File Changes

Follow these steps to modify the cable modem configuration file for SIP support.

1. Modify the [PclpClassification](#) MIB. The Classification for Upstream and Downstream packets should be set up with a different port number to indicate the correct SIP source port(s). [PclpSourcePortStart](#) and [PclpSourcePortEnd](#) should be set to **5060** in both [UpstreamPacketClassification](#) and [DownstreamPacketClassification](#).
2. (optional) Set the [arrisMtaCfgRTPDynPortStart](#) and [arrisMtaCfgRTPDynPortEnd](#) MIBs to the desired port range used for sending SIP RTP voice packets. The valid range for the start and end ports is **1024** to **65535**. The default range is **49152** to **65535**.

eDVA Configuration File Changes

Follow these steps to enable SIP support in the eDVA configuration file.

1. Modify the [ppCfgMtaCallpFeatureSwitch](#) object. To use QoS for a SIP eDVA load, the value of the feature switch must be set to **0x4020** or decimal **16416**. If these bits are not set in the feature switch, the eDVA transmits RTP packets using Best Effort. Other QoS-specific settings may be needed depending on the CMTS (and its firmware load) used.
2. Enable the eDVA device by modifying the [pktcMtaDevEnabled](#) object. The object is used to control the eDVA device; it is not line specific. The value must be set to **true** to provide dial tone on individual lines.
3. To enable individual lines, set the [ifAdminStatus](#) object for the line to **1**. The object [ifAdminStatus.9](#) corresponds to line 1, [ifAdminStatus.10](#) corresponds to line 2, and so on.
4. Add a [ppcfgMtaCountryTemplate](#) MIB setting to change tones, line card configuration, and certain Euro-PacketCable defaults (such as CallerID and VMWI) to match local requirements.

5. Enter the SIP user name by modifying the **sipCfgPortUserName** object for each line. The user name should be the phone number associated with the line. This will be used in the Caller ID string at a later date.

For line specific provisioning, **sipCfgPortUserName.1** and **.2** (and so on) represent the line numbers. The user name must be less than 255 characters. If a string contains any special characters, the entire string must be enclosed within double quotes.

6. Enter the display name by modifying the **sipCfgPortDisplayName** object. This is the display name in the SIP messages that will be used for caller name delivery at a later date. The name must be less than 255 characters. If a string contains any special characters the entire string must be enclosed within double quotes.
7. Enter the login name by modifying the **sipCfgPortLogin** object. The login name is to be specified for each device to meet the requirements for HTTP digest authentication. To provide login name privacy, this setting when read displays as "XXXXXXX" (255 Max).
8. Enter the password by modifying the **sipCfgPortPassword** object. The password is the paired requirement for the HTTP authentication on SIP messages. In order to provide password privacy, this setting when read displays as "XXXXXXX" (255 Max).
9. Enter the digit map specification by modifying the **sipCfgDigitMap** object. The digit map support on ARRIS SIP eDVA is specified in the configuration file using TLV-43 and sub-TLV69. It applies to all the lines on the Telephony Modem.

The format of the string is the same as a digit map used in MGCP (see “*Digit Map* (page 85)” for details). The maximum length for the digit map is 2048 characters.



Note 1: This digit map applies only to initial dialing. See “*Configuring Repeat Dialing* (page 104)” to set up busy indication dialing.



Note 2: TS7.1 does not support TLV-43; therefore, you cannot use TLV-43 and sub-TLV69 as in previous Touchstone firmware versions. If the digit map is less than 255 octets, you can use TLV-11.

10. To define a “busy” digit map (that is, in effect when busy tone is playing), specify the map using the **sipCfgBusyDigitMap** object. The digit map format is the same as for the default digit map, and can be set only in the configuration file.
11. If desired, modify the digit map timers:
 - T par (partial dial time-out)—modify the **pkcNcsEndPntConfigPartialDialTO** object. The default is **16** seconds.
 - T crit (critical dial timeout)—modify the **pkcNcsEndPntConfigCriticalDialTO** object. The default is **4** seconds.



Note: SIP loads support the NCS digit map timers.

12. Set the SIP call feature switch by modifying the **sipCfgSipFeatureSwitch** object. Each bit corresponds to a supported SIP call feature. See “SIP Feature Switch” for valid settings. The default value is **0**.

13. Enter the SIP provisioned CODEC by modifying the **sipCfgProvisionedCodecArray** object. This is the list of CODECs offered in the OFFER SDP. The order of this list is also the order of preference used in the OFFER SDP.
The CODEC list is a string separated by semi-colons (;). The default string is "PCMU;PCMA."
14. Enter the packetization rate by modifying the **sipCfgPacketizationRate** object. The supported packetization rate is 20 milliseconds.
15. (optional) Set the **arrisMtaCfgRTPDynPortStart** and **arrisMtaCfgRTPDynPortEnd** objects to the desired port range used for sending SIP RTP voice packets. The valid range for the start and end ports is **1024** to **65535**. The default range is **49152** to **65535**.
16. (optional) Modify the **sipCfgMaxUDPSize** object to set the maximum UDP packet size. SIP packets larger than this threshold are sent using TCP.
Default: **0**
17. Globally enable or disable calling features, if desired, by adding the **sipCfgPortFeatureSettings** object. See "Call Feature Control" for a list of calling features this object controls.
18. Enable or disable calling features on a per-line basis, if desired, by adding the **sipCfgFeatureCapability** object for the specific features. See "Per-line Call Feature Control" for a list of calling features that can be enabled or disabled on each line.

Setting up Timers

Touchstone firmware provides MIB objects for controlling the registration and retransmission (T1) timers. These objects must be set in the eDVA configuration file.



Note: The default values for these objects correspond to RFC 3261 guidelines.

1. To configure the registration wait time, set the **sipCfgRegTimerMin** and **sipCfgRegTimerMax** objects to the desired minimum and maximum wait times.
The defaults for these objects are **0** and **1800** seconds, respectively. At startup, an eDVA waits for a random amount of time (bounded by these objects) before beginning registration.
2. To change the T1 timer value, in milliseconds, set the **sipCfgT1** object. T1 is the base interval for the exponential back-off algorithm, used for retransmitting INVITE messages.
Default: **500** (ms).
3. To change the number of transmission attempts for outgoing INVITE messages, set the **sipCfgMaxRetrans** object. The default is **7**.
The defaults for this object and **sipCfgT1** provide the following retransmission timing:
 - a. Initial INVITE (0 s)
 - b. First retransmission (0.5 s) (500 ms delay)
 - c. Second retransmission (1.5 s) (1 s delay)
 - d. Third retransmission (3.5 s) (2 s delay)

- e. Fourth retransmission (7.5 s) (4 s delay)
 - f. Fifth retransmission (15.5 s) (8 s delay)
 - g. Sixth (final) retransmission (31.5 s) (16 s delay)
4. To specify a preferred session expiry timeout value, set the **sipCfgSessionExpires** object to the desired value (in seconds). When this object is set, the Telephony Modem configures the SIP INVITE message depending on the value of the object:

Value	INVITE Behavior
0	No Session-Expires header included.
1–89	The Session-Expires header includes a value of 1800.
90+	The Session-Expires header includes the value specified in the object.

The default is **1800** seconds.

Set this object to **0** to allow the proxy to control the session expiry timer.

5. To specify a suggested registration expiry timer value, set the **sipCfgRegExpires** object to the desired value. The Telephony Modem includes the specified expiry value in the REGISTER request contact header.

The actual expiry time is the lesser of the suggested value and the expiry value returned in the 200 OK response.

The default value is **0**, which omits the expiry value from the REGISTER request.

Configuring Per-Line Proxy and Registrar

Follow these steps to provision per-line proxy and registrar. Each MIB object specified must be set in the eDVA configuration file. These objects override the default proxy and registrar settings, so any line that does not specify per-line MIB objects uses the default proxy and registrar.

1. Override the default proxy address, port, and address type by setting the **sipCfgPortProxyAdr**, **sipCfgPortProxyPort**, and **sipCfgPortProxyType** objects in the eDVA configuration file; for example:

```
{ SnmpMIB sipCfgPortProxyAdr. line "host" }
{ SnmpMIB sipCfgPortProxyPort. line port }
{ SnmpMIB sipCfgPortProxyType. line type }
```

Where:

- *line* is the line number to use the override (**1** to the number of lines supported by the eDVA);
 - *host* and *port* are the IP address or FQDN, and port number of the SIP proxy;
 - *type* is the IP address type (**0** for IPv4 and **1** for DNS).
2. Override the default registrar address, port, and address type by setting the **sipCfgPortRegistrarAdr**, **sipCfgPortRegistrarPort**, and **sipCfgPortRegistrarType** objects in the eDVA configuration file; for example:

```
{ SnmpMIB sipCfgPortRegistrarAdr. line "ipaddr" }
{ SnmpMIB sipCfgPortRegistrarPort. line port }
{ SnmpMIB sipCfgPortRegistrarType. line type }
```

Where:

- *line* is the line number to use the override;
 - *ipaddr* and *port* is the IP address and port number of the SIP registrar;
 - *type* is the IP address type (**0** for IPv4 and **1** for DNS).
3. Restart the eDVA to make the per-line proxy/registrar settings take effect.

Specifying a SIP Domain Name

Follow these steps to specify a SIP domain name other than the domain specified in the proxy or provisioned FQDN.

1. Add the [sipCfgDomainOverride](#) object to the configuration file. This object must contain the desired domain name.



Note: When this object is set, the eDVA ignores the “Domain Override” SIP Feature Switch setting.

2. Restart the eDVA to download and apply the updated configuration.

Provisioning SIP Features

Use this procedure to configure end-user features with SIP loads.

Requirements and Limitations

To provision SIP features, you must first modify the CM and eDVA configuration files as described in “[Provisioning SIP Support](#) (page 91).”

You can provision up to 50 dialing features and up to 50 proxy dialing features in the eDVA configuration file.

Call Feature Control

The [sipCfgPortFeatureSettings](#) object allows you to control operation of the following call features:

- outbound Caller ID
- anonymous call rejection
- call waiting
- three-way calling

The [sipCfgPortFeatureSettings](#) object is structured as a collection of bit flags, as shown in the following table. The default value is **0**.

Bit	Description
0x40	callerIdPermanentDisable Set this bit to set the default outbound Caller ID method to “restrictive.” The default setting presents Caller ID.
0x20	anonCallRejectionEnable Set this bit to enable anonymous call rejection. The subscriber can use a “star” code to disable anonymous call rejection if desired. The default setting permits anonymous calls.
0x10	callWaitingPermanentDisable Set this bit to disable Call Waiting. The subscriber can use a “star” code to enable Call Waiting if desired. The default setting is to enable Call Waiting.
0x08	threeWayCallingDisable Set this bit to disable hook flash processing during an active call. The default setting is to allow hook flash processing.
0x04	callIdReceiptDisable Set this bit to disable local CallerID display.
0x02	callTransferEnable Set this bit to enable Call Transfer. The bit can only be set in the configuration file.

Proxy Dialing Features

Some dialing features require the eDVA to handle the tones, but the proxy handles the actual messaging. These are known as *hybrid features*. Five MIB objects control the setup and requirements for hybrid dialing features. The index number of each object groups the objects by dialing feature. These objects replace the **sipCfgDialProxyMap** object supported in loads prior to TS5.2.

sipCfgDialProxyNumber

The dialing feature number. See the table below for dialing feature values.

sipCfgDialProxyCode

A string containing the dialing code that activates the feature (for example, “*88”).

sipCfgDialProxyTone

The response tone; either **stutterTone(1)** or **di al Tone(2)**.

sipCfgDialProxyActive

A 32-bit string that identifies which lines enable the dialing feature. The least significant bit corresponds to line 1; for example, the value **3** activates the dialing feature on lines 1 and 2.

sipCfgDialProxyMessageType

For a proxy-based dialing feature, determines the type of message sent to the proxy: **i nvi te(1)** or **refer(2)**. Most dialing features should use **i nvi te(1)**.

sipCfgDialProxyMethod

(optional) Determines how the dialing code is passed to the proxy: **default**(0) prepends the dialing code to the dial string; **pc20**(1) sends the dialing code to the proxy as the host of the SIP URI and the dial string in a user parameter in the SIP URI.

Supported Dialing Features

Supported dialing features are:

Value	Feature
anonCallReject (1)	Anonymous Call Reject (ANNCJ)
anonCallRejectDisable (2)	Anonymous Call Reject Disable (ANCJD)
callForwardBusy (30)	Call Forward Busy (CALBE)
callForwardBusyDisable (31)	Call Forward Busy Disable (CALBD)
callForwardUncond (32)	Call Forward Fixed/Variable (Unconditional) (CALFV)
callForwardUncondDisable (33)	Call Forward Fixed/Variable Disable (CFFDS)
callForwardNoAnswer (34)	Call Forward No Answer (CALFN)
callForwardNoAnswerDisable (35)	Call Forward No Answer Disable (CFNAD)
warmline (36)	Warmline
warmlineDisable (37)	Disable Warmline
callReturn (50)	Call Return (CALRT)
callRedial (60)	Call Redial
callHold (61)	Call Hold
repeatDialingEnable (62)	Enable Repeat Dialing
repeatDialingCancel (63)	Cancel Repeat Dialing
callWaitTempDisable (70)	Call Waiting Temp Disable (CALWD)
callWaitPermDisableToggle (71)	Call Waiting Permanent Disable Toggle (CLWPD)
callWaitPermanentDisable (72)	Call Waiting Permanent Disable
callWaitPermanentEnable (73)	Call Waiting Permanent Enable

Value	Feature
anonCallReject (1)	Anonymous Call Reject (ANNCJ)
callerIDPermBlockToggle (90)	Caller ID Permanent Block Toggle (CIDPB)
callerIDTempEnable (91)	Caller ID Temp Enable (CIDTE)
callerIDTempBlock (92)	Caller ID Temp Block (CIDTB)

Action

Perform the following tasks as necessary:

- [Setting up Dialing Features](#)..... 102
- [Configuring Warmline or Hotline](#) 103
- [Configuring Repeat Dialing](#) 104
- [Configuring T.38 and Fax-Only Modes](#) 105
- [Configuring Distinctive Ring/Alert Tones](#) 105

Setting up Dialing Features

Add the MIB objects described below to the eDVA configuration file to set up dialing features. The **sipCfgDialFeatMap** MIB object, provided in earlier loads, is supported for backwards compatibility but no longer documented. The MIB objects for each feature are distinguished by the index; for example, **sipCfgDialFeatName.1** and **sipCfgDialFeatName.2** are two different features.



Note: Certain dialing features, including Hotline, Warmline, and Repeat Dialing, require further configuration. See the appropriate task for any extended configuration required.

1. Add the **sipCfgDialFeatName** object to enable particular dialing features. See “Supported Dialing Features” for a list of supported features.
2. Add the **sipCfgDialFeatCode** object to define a dialing code for a particular feature. You can specify up to three codes, separated by a comma, for each feature. For example, use ***70,1170** to allow either ***70** or **1170** to activate a feature.
3. Add the **sipCfgDialFeatTone** object to specify the confirmation tone used when activating a feature. The choices are **stutterTone** (default) and **dialTone**.
4. Add the **sipCfgDialFeatActive** object to assign a feature to one or more lines on the eDVA. The value for this object is a bit mask; each bit represents one line. The least significant bit corresponds to line 1.

Examples: A value of **0. 0. 0. 3** enables the dialing feature for line 1 and line 2. A value of **0. 0. 0. 2** enables the feature only for line 2.

- Set the **sipCfgDialFeatMode** object to enable the feature for the appropriate dialing phases. This object is a set of bits:

Bit Value	Description
0x02	Busy
0x01	Initial dialing

These bits can be combined; a value of **3** allows the dialing code to be used both during initial dial tone and during a busy signal. The default is **1**.

- For hybrid features—those features where the eDVA handles the tones, but the proxy handles the actual messaging—add the objects described in “Proxy Dialing Features.”

Configuring Warmline or Hotline

The SIP load includes support for specifying a hotline or warmline number in the configuration file. A hotline automatically dials the specified number as soon as the specified line goes off-hook; a warmline automatically dials the specified number after providing dial tone for a specified amount of time. Add the objects described below to the eDVA configuration file to set up a hotline or warmline for a specific line.

- In the configuration file, set the **sipCfgPortWarmOrHotlineNumber** object to the phone number to dial. This object is specific to a line, so it must be specified with the line number; for example, **sipCfgPortWarmOrHotlineNumber.2** for line 2.
- In the configuration file, set the **sipCfgPortWarmLineTimeout** object to the timeout value (in seconds) for a warmline, or to **0** for a hotline. This object is specific to a line, so it must be specified with the line number; for example, **sipCfgPortWarmLineTimeout.1** for line 1.
- To allow the subscriber to specify a warmline number, add an entry to the **sipCfgDialFeatTable** in the configuration file. The following example specifies ***53** and ***54** as the feature codes to enable and disable warmline dialing, enabling the feature on lines 1 and 2.

```

SnmpMib = sipCfgDialFeatName.15 warmline
SnmpMib = sipCfgDialFeatCode.15 "*53"
SnmpMib = sipCfgDialFeatTone.15 stutterTone
SnmpMib = sipCfgDialFeatActive.15 hexstr: 0.0.0.3
SnmpMib = sipCfgDialFeatName.16 warmlineDisable
SnmpMib = sipCfgDialFeatCode.16 "*54"
SnmpMib = sipCfgDialFeatTone.16 stutterTone
SnmpMib = sipCfgDialFeatActive.16 hexstr: 0.0.0.3

```

The **sipCfgDialFeatActive** object specifies the lines on which the specified feature is active. It consists of a series of bits, with the least significant bit corresponding to line 1. Other effects are:

- Setting the warmline feature code uses the value set in the **sipCfgPortWarmLineTimeout** MIB object for the timeout. If the object is not set for a line in the configuration file, the default is 5 seconds.

- A subscriber-specified warmline number replaces the value for **sipCfgPortWarmOrHotlineNumber** that was specified in the configuration file.
- The **sipCfgPortWarmOrHotlineNumber** MIB object, whether set in the configuration file or by the subscriber, shows the currently-configured warmline number. If the subscriber disables warmline, This object contains a 0-length string.
- If you do not configure warmline feature codes, the number specified in **sipCfgPortWarmOrHotlineNumber** is permanent and cannot be changed or disabled by the subscriber.

Configuring Repeat Dialing

Follow these steps to configure the Repeat Dialing feature. Configuring any busy indication dialing feature uses the same steps.

1. Add two entries to the **sipCfgDialFeatTable** in the eDVA configuration file, to set up and cancel Repeat Dialing. The following example enables Repeat Dialing when the subscriber presses ***5** during a busy signal, and cancels Repeat Dialing with **#5**:

```
/* enable Repeat Dialing */
{Snmplib sipCfgDialFeatName.1 Integer 62}
{Snmplib sipCfgDialFeatCode.1 String "*5"}
{Snmplib sipCfgDialFeatTone.1 Integer 1} /* stutter */
{Snmplib sipCfgDialFeatActive.1 HexString 0x00000003}
{Snmplib sipCfgDialFeatMode.1 Integer 2} /* busy mode only */
/* cancel Repeat Dialing */
{Snmplib sipCfgDialFeatName.2 Integer 63}
{Snmplib sipCfgDialFeatCode.2 String "#5"}
{Snmplib sipCfgDialFeatTone.2 Integer 1} /* stutter */
{Snmplib sipCfgDialFeatActive.2 HexString 0x00000003}
/* Repeat Dialing timer settings */
{Snmplib sipCfgRepeatDialingInterval.0 10}
{Snmplib sipCfgRepeatDialingTimeout.0 30}
{Snmplib sipCfgRepeatDialingSessionProgressTimer.0 60}
```

2. Add the **sipCfgBusyDigitMap** MIB object to the eDVA configuration file, specifying the strings that can be matched while the eDVA is playing a busy tone:

```
Snmplib = sipCfgBusyDigitMap.0 String "*x|#x";
```

3. Add the following MIB objects to the eDVA configuration file to set related timers:

sipCfgRepeatDialingInterval

The time, in seconds, between repeat dialing attempts. Default: **30** seconds.

sipCfgRepeatDialingTimeout

The time, in seconds, that Repeat Dialing is active (and unsuccessful) before the eDVA cancels the feature. Default: **1800** seconds.

sipCfgRepeatDialingSessionProgressTimer

The time, in seconds, the eDVA waits after receiving a "183 Session Progress" provisional response before alerting the subscriber. This delay is needed because many PSTN calls receive this response before receiving a negative INVITE. Default: **2** seconds.



Note: This object must be set in the configuration file. Changes made to this object persist across reboots.

4. Reset the eDVA to enable the feature.

Configuring T.38 and Fax-Only Modes

Follow these steps to configure T.38 and fax-only modes for a line. Each MIB object specified must be set in the eDVA configuration file.

1. Set the T.38 mode by adding the **sipCfgPortT38Mode** MIB object to the eDVA configuration file. This MIB object is specific to a line, so it must be specified with the line number; for example, **sipCfgPortT38Mode.2** for line 2.

Valid settings are: **t380ff**(1), **t38Loose**(2), and **t38Strict**(3). The default is **t380ff**.

2. Set fax-only mode by adding the **sipCfgPortFaxOnlyTimeout** MIB object to the eDVA configuration file. This MIB object is specific to a line, so it must be specified with the line number; for example, **sipCfgPortFaxOnlyTimeout.1** for line 1. The value specifies the timeout, in seconds, after which the eDVA drops the call if it does not detect fax or modem tones.

Valid range: **0** (disabled) to **600** seconds. The default is **0**.

Configuring Distinctive Ring/Alert Tones

Follow these steps to set the expected Alert-Info strings for distinctive ringing and alert (Call Waiting) tones. Each MIB object specified must be set in the eDVA configuration file. For more information, see [Distinctive Ringing](#) (page 82).

1. In the eDVA configuration file, set the following MIB objects:

MIB Object	Description
sipCfgAlertInfoR0	The value of the Alert-Info header field to instruct the eDVA to play the R0 ring cadence.
sipCfgAlertInfoR1	The value of the Alert-Info header field to instruct the eDVA to play either the R1 ring cadence (call not active) or the WT1 call waiting tone (call active).
sipCfgAlertInfoR2	The value of the Alert-Info header field to instruct the eDVA to play either the R2 ring cadence (call not active) or the WT2 call waiting tone (call active).
sipCfgAlertInfoR3	The value of the Alert-Info header field to instruct the eDVA to play either the R3 ring cadence (call not active) or the WT3 call waiting tone (call active).
sipCfgAlertInfoR4	The value of the Alert-Info header field to instruct the eDVA to play either the R4 ring cadence (call not active) or the WT4 call waiting tone (call active).

MIB Object	Description
sipCfgAlertInfoR5	The value of the Alert-Info header field to instruct the eDVA to play the R5 ring cadence.
sipCfgAlertInfoR6	The value of the Alert-Info header field to instruct the eDVA to play the R6 ring cadence.
sipCfgAlertInfoR7	The value of the Alert-Info header field to instruct the eDVA to play the R7 ring cadence.

The default value for each MIB object is `<file: //Bellcore/rx>`, where *x* is the ring cadence (0 through 7). You may need to change these if the P-CSCF sends different Alert-Info strings.

- To configure distinctive ringing to alert the subscriber to a call on hold when the line is on hook, set the **sipCfgDistinctiveRingingForCallHold** object. The allowed values are:
 - standard**(0) (default) — use the standard ring cadence.
 - r0**(1) — use the R0 ring cadence.
 - r1**(2) — use the R1 ring cadence.
 - r2**(3) — use the R2 ring cadence.
 - r3**(4) — use the R3 ring cadence.
 - r4**(5) — use the R4 ring cadence.
 - r5**(6) — use the R5 ring cadence.
 - r6**(7) — use the R6 ring cadence.
 - r7**(8) — use the R7 ring cadence.

Provisioning PacketCable 2.0 SIP Loads

This chapter applies only to SIP PC20 loads.

PacketCable 2.0 Concepts

This section describes PacketCable 2.0 terminology and concepts.

Terminology

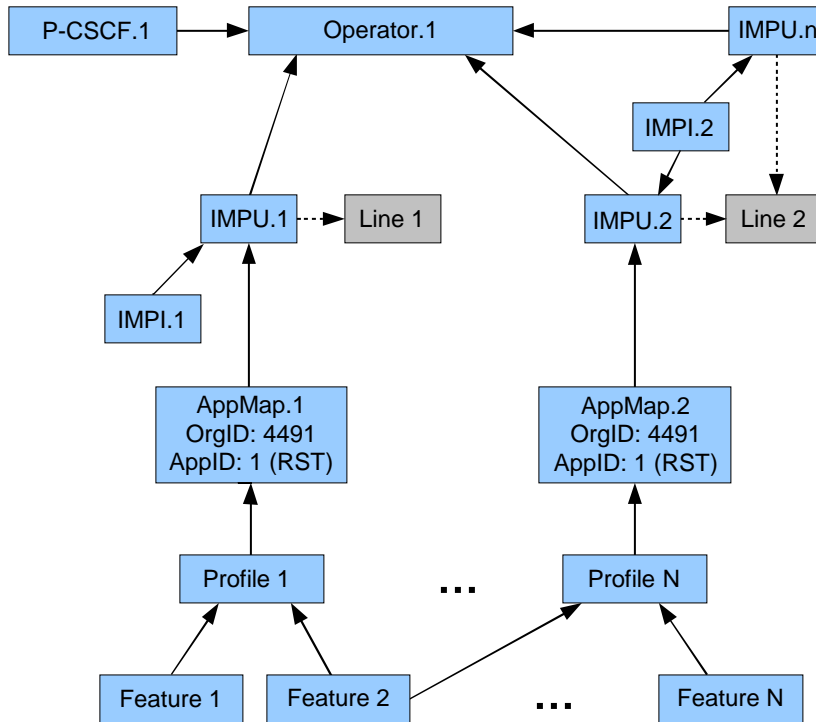
PacketCable 2.0 introduces new terminology and abbreviations for telephony components within a network. The following table shows PacketCable 1.x terms with their equivalent PacketCable 2.0 terms.

PC1.x	PC2.0	Description
eMTA	E-UE, E-DVA	A device consisting of a combined DOCSIS CM and a PacketCable eDVA (MTA).
MTA	eUE, eDVA	The logical PacketCable device.
Proxy	P-CSCF	Proxy-Call Session Control Function. In PacketCable 2.0, the P-CSCF communicates with the registrar, so it is no longer necessary to specify a registrar in eDVA configuration.

This document uses the terms *E-UE* for the entire Telephony Modem, *eDVA* for the telephony portion, and *CM* for the cable modem portion.

Configuration Concepts

Provisioning an eDVA conforms to the following model. Note that AR01.1 uses a simplified version of the PacketCable 2.0 model, and the simplified version is shown as follows:



The PacketCable 2.0 Residential SIP Telephony (RST) model is an abstract representation of the capabilities and features available on a phone line. The intent is to provide flexibility for future services. The RST specification introduces the following terms:

Operator

The operator indirectly associates users with a P-CSCF. Multiple operators could be associated with various third-party telephony providers or with remote locations in the MSO's network. AR01.1 supports one operator definition.

User

A user consists of two components:

- **IMPU (IP Multimedia Public User)** — the public identity of the user. This could be the phone number assigned to the user, or some other unique user name. The IMPU associates a user with a phone line through the [pktcEUEUsrIMPUAdditionalInfo](#) object. Multiple users may share a line, implementing a Teen Line service or as part of a transition to a new area code or exchange.
- **IMPI (IP Multimedia Private Identity)** — the private identity of the user. Primarily, the IMPI defines the credentials for the IMPU.

Application

Defines the service(s) to be provided to a user. AR01.1 supports only the Residential SIP Telephony (RST) application. The application map defines the application and selects a profile.

Profile

Selects a digit map and an associated collection of features. A profile may be applied to one or more users.

Feature

Defines capabilities and dialing features available to a profile. Features include basic call functionality (including timer durations), announcements, Caller ID display, and common dialing features.

Each feature is defined as an entry in one or more tables, allowing profiles to contain different feature configurations. User-based features are indexed normally; network-based features (or network-based components of features) use the Operator index.

The following restrictions apply:

- Touchstone firmware supports only one P-CSCF (but can fail over to a second P-CSCF) and only one operator per E-UE.
- Touchstone firmware supports only the PacketCable Residential SIP Telephony (RST) application.
- Multiple users can be assigned to a line, but a user can be associated only with one line.

Each box in the above figure is associated with a table in the PacketCable 2.0 MIBs. Arrows represent an index to a row in each table, connected with another table. The boxes labeled IMPU and IMPI are contiguous since AR01.1 requires them to be permanently associated.

Supported Features

The following table shows which RST feature MIB tables are supported in AR01.1.

RST Feature	MIB Table Support	
	Network	User
Basic Call	Partial	Partial
Announcement	Partial	No
Activation Status	n/a	Partial
No Answer Timeout	n/a	Partial
CallerID	n/a	Partial
CallerID Display	n/a	Partial
CallerID Block	n/a	No
CallerID Delivery	n/a	No
Call Forward	No	Partial
Call Hold	n/a	No
Call Transfer	n/a	No

RST Feature	MIB Table Support	
	Network	User
Basic Call	Partial	Partial
Do Not Disturb	n/a	No
MWI	Partial	n/a
Auto Recall	n/a	No
Auto Callback	n/a	No
Busy Line Verify	No	n/a
Emergency Services	Partial	n/a
Selective Call Forward	n/a	No

For details about partially supported feature tables, see "[Standards Compliance](#) (page 16).

DHCP Option 60 Support

AR01.1 uses DHCP option 60 in DHCP Discover messages to advertise PacketCable support. The option contains the string "pktc2.0" to indicate PacketCable 2.0 support.

The following table lists the sub-options sent with DHCP option 60 and their meanings:

Sub-option	Value	Description
1	0x02	PacketCable 2.0
2	0x02	Number of endpoints (2)
3	0x00	TGT support (no)
4	0x00	HTTP download file access method (no)
5	0x01	MTA-24 Event Syslog notification
9	0x01	NVRAM Ticket/Session Key Storage supported
10	0x01	Provisioning Event Reporting supported
11	0x0106090f	Supported CODECs:
		01 = other
		06 = PCMU
		09 = PCMA
		0F = telephone-event
13	0x01	Echo Cancellation supported
16	0x09	ifIndex of first phone line

Sub-option	Value	Description
18	0x0007	Supported provisioning flows: Secure, Hybrid, Basic
19	0x01	T.38 Version support (v0)
20	0x01	T.38 Error Correction support (redundancy)
21	0x01	RFC 2833 DTMF relay supported
22	0x01	Voice Metrics supported
23	0x02001f	CableLabs MIBs supported: EN-SIG, EN-MTA, EVENT-MIB, SIG-MIB, MTA-MIB
24	0x00	Multiple Grants per Interval (no)
25	0x00	V.152 support (no)
26	0x00	Certificate Bootstrapping (no)

Overview of SIP Features

This section describes some important features of the SIP loads.

Barge-In

The `Join` header (RFC 3911) may be used to barge-in to an existing call. A brief tone is played to the existing call as the calls are conferenced together.

Loopback

The SIP eDVA can terminate loopback calls. When a loopback call is received, the phone does not ring. It automatically answers the call and loops back media or packets to the originator. The eDVA has a 2 call per line resource limitation. The user may make a call or receive a call while a loopback call is in progress. If another call is placed, either by an incoming call or a new outbound call, the loopback call is disconnected.

Packet Loopback is analogous to NETWLOOP, and controlled by the `sipCfgPacketLoopbackNumber` object. Media Loopback is analogous to NETWTEST, and controlled by the `sipCfgMediaLoopbackNumber` object.

The *Loopback Draft* (<http://www.ietf.org/internet-drafts/draft-ietf-mmusic-media-loopback-08.txt>) specifies the CODEC negotiation involved in setting up a loopback call.

Touchstone firmware also supports a proprietary loopback method. If the eDVA receives a call from a number matching those provisioned in specific MIB objects, it creates a loopback call.

Extended Offhook Processing

Touchstone firmware partially implements extended offhook processing as defined in PKT-SP-RSTF-I08-110127, section 7.1.4.4. Touchstone firmware implements extended offhook processing as shown below.

Origination Mode

Origination mode typically involves the subscriber picking up the phone, then either not dialing a number or pausing too long between digits. PacketCable defines an Origination Mode Dial Time timer and a Long Interdigit Timer (typically 16 seconds each); either timer expiring invokes the “permanent sequence.”

Use the **pktcEUERSTNfBCallOrigDTTimer** object to set the Dial Time timer. The default time is 16 seconds.

Use the **pktcEUERSTNfBCallOrigModLongIntDig** object to set the Long Interdigit timer. The default time is 16 seconds.

Termination Mode

When the eDVA receives the BYE signal, it starts a timer. Prior to TS7.6 MSUP2, the timer is fixed at 20 seconds. In TS7.6 MSUP2 and newer loads, the **sipCfgTermOffHookProcessingDelay** object controls the timer. The valid range for the timer is **0** to **60** seconds; the default is **20** seconds.

If the timer expires before the subscriber hangs up, the eDVA invokes the “permanent sequence.”

Permanent Sequence

A series of MIB objects, defined in the CL-PKTC-EUE-RST-MIB, specify the permanent sequence played in response to an extended offhook. The default behavior is:

D11PLUS loads:

1. Reorder tone for 30 seconds
2. OSI for 1 second
3. Silence for 4 seconds
4. Howler tone for 60 seconds

Other loads:

1. OSI for 1 second
2. Silence for 10 seconds
3. Howler tone for 60 seconds

If the subscriber has not hung up the phone by the time the sequence ends, the line enters a lockout state until the subscriber hangs up.

The following table shows the MIB objects used to provision the permanent sequence and the default values for each object.

Object	.TW Loads	Other Loads
pkcEUESTNfBCallPermSeqTone1	"file:///PacketCableRST/ro"	"file:///PacketCableRST/osi"
pkcEUESTNfBCallPermSeqTimer1	30	1
pkcEUESTNfBCallPermSeqTone2	"file:///PacketCableRST/osi"	"file:///PacketCableRST/nt"
pkcEUESTNfBCallPermSeqTimer2	1	10
pkcEUESTNfBCallPermSeqTone3	"file:///PacketCableRST/nt"	"file:///PacketCableRST/ot"
pkcEUESTNfBCallPermSeqTimer3	4	60
pkcEUESTNfBCallPermSeqTone4	"file:///PacketCableRST/ot"	""
pkcEUESTNfBCallPermSeqTimer4	60	0

Emergency Calls

Any number matching the pattern associated with the digit map action **EMERGENCY- CALL** is treated as an emergency call. Emergency calls have special treatment. The subscriber is, by default, not allowed to terminate an emergency call (this can be overridden by setting the **sipCfgSipFeatureSwitch** bit **0x00000100**). If the user goes on hook, and the default behavior is in effect, the eDVA does not send a BYE message. Instead, the call is put on "Network Hold" as follows:

1. When the subscriber goes on-hook during an emergency call, the Touchstone eDVA sends an INVITE (priority: emergency SDP: a=inactive). This causes the PSAP operator to hear a tone that indicates that the user went on-hook.
2. The PSAP operator can send an INVITE (priority: emergency SDP: a=sendrecv) that causes the eDVA to ring. If the subscriber goes off-hook before receiving such an INVITE, then it should send the invite to re-establish two-way communications.
3. The Network Hold Timer specifies the maximum time that an emergency call is preserved in the Network Hold state. The timer is started every time that the user goes on-hook during an emergency call, and is cleared if the user goes off hook. If the Network Hold Timer expires, then the eDVA sends a BYE message to finally terminate the call. The default value for the Network Hold Timer is 45 minutes.

Emergency calls have the following restrictions:

- Outgoing emergency calls offer only the G.711 CODEC in the SDP.
- Caller ID blocking is overridden on outgoing emergency calls.
- In the INVITE, the eDVA does not include the route received in the Service-Route: header that was received in the registration response.
- Incoming emergency calls cannot be transferred.

Incoming calls are treated as emergency calls under either of the following conditions:

- The P-Asserted-Identity header value matches the value of the **sipCfgEmergencyServiceURN** object (the default is URN: service: sos). If you change this value, always begin the string with the **URN:** prefix to comply with RFC 5031.
- The Priority header value is “emergency.”

To disconnect any active call when originating or receiving an emergency call, set bit **0x00001000** in the **sipCfgSipFeatureSwitch**.

Distinctive Ringing

Several distinctive ringing types, corresponding to R0 through R7, are defined by the country template. If the Alert-Info header is received, Touchstone firmware compares the header to the strings specified in the MIB objects **sipCfgAlertInfoR0** through **sipCfgAlertInfoR7**. If the header matches one of these, the eDVA plays the corresponding R0–R7 tone. AR01.1 also supports tones WT1 through WT4 for use with Call Waiting, and maps R1 through R4 to WT1 through WT4 for a Call Waiting alert.

The following configuration shows the default string settings.

```
{ SnmpMib sipCfgAlertInfoR0.0 "< http://127.0.0.1/Bellcore-dr0 >" }
{ SnmpMib sipCfgAlertInfoR1.0 "< http://127.0.0.1/Bellcore-dr1 >" }
{ SnmpMib sipCfgAlertInfoR2.0 "< http://127.0.0.1/Bellcore-dr2 >" }
{ SnmpMib sipCfgAlertInfoR3.0 "< http://127.0.0.1/Bellcore-dr3 >" }
{ SnmpMib sipCfgAlertInfoR4.0 "< http://127.0.0.1/Bellcore-dr4 >" }
{ SnmpMib sipCfgAlertInfoR5.0 "< http://127.0.0.1/Bellcore-dr5 >" }
{ SnmpMib sipCfgAlertInfoR6.0 "< http://127.0.0.1/Bellcore-dr6 >" }
{ SnmpMib sipCfgAlertInfoR7.0 "< http://127.0.0.1/Bellcore-dr7 >" }
```

Touchstone firmware supports both the Bellcore-defined tones shown in the listing above, and the equivalent PacketCable 2.0-defined tones **file:///PacketCableRST/rn**, where *n* is 0 through 7.

Configuring PacketCable 2.0 SIP

Use this procedure to configure PacketCable 2.0 SIP for Touchstone products.

Configuration Overview

The most efficient way to configure PacketCable 2.0 SIP is to use a “top-down/bottom-up” approach, building and interlinking the MIB tables in the following order:

1. Operator table
2. P-CSCF table
3. Feature and Digit Map tables
4. Profile to Feature tables
5. Application Map tables
6. IMPI table
7. IMPU table

See *PacketCable 2.0 Concepts* (page 107) for an overview of how the tables interconnect.

Action

Perform the following tasks in the order shown.

[1] Configuring Operator Information	115
[2] Configuring Users and Features	116
[3] Configuring Extended Offhook Processing	116
[4] Post-Provisioning SIP Lines	117
[5] Configuring T.38 and Fax-Only Modes	118

Configuring Operator Information

Operator information includes the Operator and P-CSCF tables. Follow these steps:

1. Configure the Operator table. The index for each object is typically **1**; this index is used in other tables to refer to this Operator table.
 - a. Set the **pktcEUEDevOpDomain** object to the domain name; for example, `atl.ga.example.com`.
 - b. Set the **pktcEUErstKeepAliveSetting** object to set the in-service/out-of-service status:
 - Set to **on**(1) to use the keep-alive mechanism to determine the status.
 - Set to **off**(2) to use the RSTF in-service state requirements.
 - Set to **conditional** (3) to set the keep-alive mechanism depending on the requirements in the 200 OK response to the REGISTER message.
 - c. Set the **pktcEUEDevOpRowStatus** object to **createAndGo**.
2. Configure the P-CSCF table. The index for each object is **.oper.1**, where **oper** is the index of the operator table (usually 1).
 - a. Configure the P-CSCF address:

pktcEUEDevPCSCFAddrType
Either **ipv4** or **ipv6** to specify the IP address type, or **dns** to specify a fully-qualified domain name.

pktcEUEDevPCSCFAddr
The address of the P-CSCF associated with the operator.

pktcEUEDevPCSCFSipPort
(optional) The SIP port. Default: **5060**.
 - b. (optional) If desired, modify the following SIP timers:

pktcEUEDevPCSCFTimerT1SIP Timer T1. Default: **500** ms.**pktcEUEDevPCSCFTimerT2**SIP Timer T2. Default: **4000** ms.**pktcEUEDevPCSCFTimerT4**SIP Timer T4. Default: **5000** ms.**pktcEUEDevPCSCFTimerTD**SIP Timer TD. Default: **32000** ms.

3. Set the **pktcEUEDevPCSCFRowStatus** object to **createAndGo**.

Configuring Users and Features

Follow these steps to configure user and feature tables:

1. To configure features, proceed to [Provisioning PacketCable 2.0 Features](#) (page 118).
2. To configure user information, proceed to [Provisioning PacketCable 2.0 Users](#) (page 130).
3. To configure application profiles, proceed to [Provisioning PacketCable 2.0 Application Profiles](#) (page 132). Digit map configuration is described in [Configuring PacketCable 2.0 Digit Maps](#) (page 124).
4. To configure the Application Map, proceed to [Provisioning PacketCable 2.0 Application Maps](#) (page 133).

Configuring Extended Offhook Processing

Follow these steps to provision extended offhook processing.

1. Provision the origination mode timers (Origination Mode Dial Time and Long Interdigit Timer) in one of the two following ways:
 - a. In the digit map, configure the **T** (Dial Time) and **L** (Long Interdigit Timer) timers.
 - b. Configure the following two MIB objects:
 - **pktcEUESTNfBCallOrigDTTimer**
 - **pktcEUESTNfBCallOrigModLongIntDig**

The default value for both timers is 16 seconds.



Note: Use either the digit map or the MIB objects to configure these timers; do not use both.

2. To set the sequence of tones played, and the duration of each, set the following MIB objects:

pktcEUESTNfBCallPermSeqTone1

The URI of tone 1. Example: file:///PacketCableRST/ro

pktcEUESTNfBCallPermSeqTimer1

The duration, in seconds, for tone 1.

pktcEUERSTNfBCallPermSeqTone2

The URI of tone 2.

pktcEUERSTNfBCallPermSeqTimer2

The duration, in seconds, for tone 2.

pktcEUERSTNfBCallPermSeqTone3

The URI of tone 3.

pktcEUERSTNfBCallPermSeqTimer3

The duration, in seconds, for tone 3.



Note: The **pktcEUERSTNfBCallPermSeqTone4** and **pktcEUERSTNfBCallPermSeqTimer4** objects are not supported.

3. (optional) Configure the offhook processing delay time (in seconds) by setting the **sipCfgTermOffHookProcessingDelay** object. Valid range: **0** to **60** seconds. Default: **20** seconds.

Post-Provisioning SIP Lines

Follow these steps to change the provisioning status of one or more lines without rebooting the eDVA.



Note: This task applies only to line-level parameters.

The following MIB objects require a restart of the SIP stack and are not updated by this feature:

- **sipCfgGenLinger**
- **sipCfgTimerF**

1. Create a configuration file containing the line-level parameters to change, and make the file available for download on a reachable TFTP server.
2. Using an SNMP network manager, set the **arrisMtaDevSipConfigFileURL** object to the URL of the configuration file.
3. Using an SNMP network manager, set the **arrisMtaDevSipDwnldConfig** object to **on(2)**.
The eDVA downloads the configuration file and applies the line-level changes.
Touchstone firmware does not take lines out of service or interrupt calls in process.
4. To make the configuration changes permanent, modify the standard configuration file to reflect the changes.



Note: If the post-provisioning changes are not copied to the standard configuration file, the eDVA loses those changes when re-initialized.

Configuring T.38 and Fax-Only Modes

Configuring T.38 and fax operation is identical for both PC20 and ARRIS SIP loads. See [Configuring SIP T.38 and Fax-Only Modes](#) for details.

Provisioning PacketCable 2.0 Features

Use this procedure to provision individual PacketCable 2.0 features.

Feature Support

In PacketCable 2.0, a “feature” includes anything that can be enabled or configured. If all features are disabled, the eDVA can receive incoming calls and make only emergency outgoing calls. By default, all features are enabled and have reasonable default provisioning.

All features are defined and enabled in the [pktcEUERSTAppProfileToFeatTable](#). Some features may be further configured as follows:

digit map

In addition to defining valid phone numbers, the digit map defines Vertical Service Codes used to invoke or control dialing features.

Network table

Network tables are associated with an operator table, and control provisioning associated with all users. Not all features have network tables.

User table

User tables are associated with individual users, and control provisioning for that user. Not all features have user tables.

The [pktcEUERSTAppFeatID](#) object specifies a feature type. The following are valid features:

- **digitMap**(2) — A digit map that enables the supported features.
- **basicCall**(3) — Basic Call capabilities. The user-side table defines the CODECs available; the network-side table defines timers and the off-hook alert sequence.
- **announcement**(4) — Announcements. The Announcement Map table defines tones to play when the eDVA receives various response codes (such as “486 Busy”).
- **statusChange**(5) — UE Status Change. This feature defines the minimum registration expiration interval.
- **noAnswerTimeout**(6) — Defines how long the eDVA rings a line before sending a “480 Temporarily Unavailable” response to the originator.
- **callerID**(7) — Sets the preferred presentation status (anonymous or public) on Caller ID for outgoing calls.
- **callerIDDisplay**(8) — Controls the display of Caller ID information for incoming calls.
- **callerIDBlocking**(9) — Controls per-call Caller ID Blocking. This feature is configured entirely through the digit map. No other provisioning is supported.
- **callerIDDelivery**(10) — Controls per-call Caller ID Delivery. This feature is configured entirely through the digit map. No other provisioning is supported.

- **callForwarding(11)** — Controls audible indicators for Call Forwarding (CFV). The network table controls the reminder dialtone when CFV is active; the user table controls Ring Reminder.
- **callWaiting(12)** — The Call Waiting feature is configured entirely through the digit map. Beyond enabling and disabling the feature, no other provisioning is supported.
- **callHold(13)** — The Call Hold feature is configured entirely through the digit map. Beyond enabling and disabling the feature, no other provisioning is supported.
- **callTransfer(14)** — Call Transfer is supported in the digit map. Setting the **pkcEuerstCIncomingOnly** object to **true(1)** allows Call Transfer only when the current call (first call leg) is an incoming call (this is the default setting for .TW loads).
- **threeWayCalling(15)** — 3-Way Calling is enabled in the digit map. and in the ARRIS-proprietary **arrisSipMib**.
- **doNotDisturb(16)** — Do Not Disturb is enabled in the digit map. The **pkcEuerstDnDTable** configures confirmation tones for enabling and disabling the feature.
- **subscrProgPin(17)** — Not supported.
- **msgWaitIndicator(18)** — Message Waiting Indicator (MWI) subscription duration is configured through a network table. If the MWI Application Server (MWI AS) is different from the P-CSCF, the ARRIS-proprietary **arrisSipMib** can be used to specify the address of the MWI AS, and to control the behavior of MWI.
- **autoRecall(19)** — Auto Recall is supported through the digit map. The **pkcEuerstAutoRclTable** is not supported in AR01.1.
- **autoCallback(20)** — Auto Callback is supported through the digit map. The **pkcEuerstAutoCbTable** is not supported in AR01.1.
- **busyLineVerify(21)** — Not supported.
- **emergencySvc(22)** — Emergency Service timers are configured through the **pkcEuerstNfEmSvcTable**. The eDVA fills emergency service-related **arrisSipMib** objects with provisioned or server-provided information.
- **scf(23)** — Selective Call Forwarding is not supported.
- **acr(24)** — Anonymous Call Rejection is configured entirely through the digit map. No other provisioning is supported.
- **solicitorBlocking(25)** — Solicitor Blocking is configured entirely through the digit map. No other provisioning is supported.
- **distinctAlerting(26)** — Distinctive Alerting is configured entirely through the digit map. No other provisioning is supported.
- **speedDialing(27)** — Not supported.
- **cot(28)** — Customer Originated Call Trace (COT) is configured entirely through the digit map. No other provisioning is supported. Use the CallP **cotdump** CLI command to display information captured by the eDVA in response to the subscriber using the COT dialing code.
- **heldMedia(29)** — Held Media. Enables an active bi-directional media stream to be held. No configuration is required or supported for this feature.

- **hotline(31)** — Hotline/Warmline. When the subscriber goes off-hook and does not dial a digit before the **pktcEUSERSTHotlineOffhookTimer** expires, the eDVA automatically dials a number. By setting the timeout to 0, this feature acts as a hotline, automatically dialing the designated number as soon as the subscriber goes off-hook.



Note: The ARRIS-proprietary MIB objects **sipCfgWarmOrHotlineNumber**, **sipCfgWarmLineTimeout**, and **sipCfgPortWarmOrHotlineEnable** may also be used to provision hotline or warmline. If the **pktcEUSERSTHotlineDestAddress** object is not provisioned, the eDVA uses the ARRIS SIP objects.

In addition to the PacketCable 2.0 features described above, Touchstone firmware provides the following ARRIS-specific features.

- **Warmline/Hotline** — when the subscriber goes off-hook and does not dial a digit before the **sipCfgPortWarmLineTimeout** timer expires, the eDVA automatically dials the number specified by the **sipCfgPortWarmOrHotlineNumber** object. By setting the timeout to 0, this feature acts as a hotline, automatically dialing the designated number as soon as the subscriber goes off-hook.



Note: AR01.1 does not support dialing codes for subscriber control over the Warmline/Hotline feature.

- **3-way Calling** — An ARRIS extension to the PacketCable 2.0 digit map provides subscriber control over 3-way calling. Two actions, 3WC-ENABLE and 3WC-DISABLE, allow provisioning dialing codes for use with 3-way calling.

P-CSCF Dialing Features

In AR01.1, any dialing features not supported by the digit map are treated as P-CSCF dialing features. To enable a P-CSCF dialing feature, map the P-CSCF-defined dialing code to the **MAKE- CALL** action, passing the dialing code to the P-CSCF.

Example:

```
"*99" : MAKE- CALL ( "sip: " #0 =domain =dialString )
```

The P-CSCF must instruct the eDVA to play stutter tone or provide other confirmation, if necessary.

Action

Perform the following tasks as needed.

- [Basic Call Configuration](#) 121
- [Configuring the Status Change Feature](#)..... 122
- [Configuring No Answer Timeout](#) 122
- [Configuring Caller ID](#) 122
- [Configuring Emergency Services](#) 123
- [Configuring Distinctive Ring/Alert Tones](#) 124

Basic Call Configuration

Follow these steps to configure a basic call feature.

1. Configure the user-side table, [pktcEUERSTBasicCallTable](#), as follows:

[pktcEUERSTBCallPrefCodecList](#)

(optional) A comma-delimited list of the CODECs sent in the SDP. If left unconfigured, the eDVA offers G.711.



Note: For backward-compatibility with earlier versions of Touchstone firmware, this object also accepts a semicolon-delimited list supported by the now-deprecated [sipCfgProvisionedCodecArray](#) object.

[pktcEUERSTBCallStatus](#)

Set to **createAndGo(4)**.

2. (optional) Configure the network-side table, [pktcEUERSTNfBasicCallTable](#), as follows. The index is *.oper.x*, where *oper* is the index of the entry in the Operator table. All the values in this table have reasonable defaults.

[pktcEUERSTNfBCallByeDelay](#)

The Bye delay, in seconds. When set to a non-zero value, a called party can hang up the phone then pick up before the timer expires without ending the call.

[pktcEUERSTNfBCallPermSeqTone1](#)

The URI specifying the first tone in the permanent sequence. See [Permanent Sequence](#) (page 80) for details and defaults.

[pktcEUERSTNfBCallPermSeqTimer1](#)

The duration of the first tone in the permanent sequence.

pktcEUERSTNfBCallPermSeqTone2

The URI specifying the second tone in the permanent sequence.

pktcEUERSTNfBCallPermSeqTimer2

The duration of the second tone in the permanent sequence.

pktcEUERSTNfBCallPermSeqTone3

The URI specifying the third tone in the permanent sequence.

pktcEUERSTNfBCallPermSeqTimer3

The duration of the third tone in the permanent sequence.

pktcEUERSTNfBCallPermSeqTone4

The URI specifying the last tone in the permanent sequence.

pktcEUERSTNfBCallPermSeqTimer4

The duration of the last tone in the permanent sequence.

pktcEUERSTNfBCallOrigModLongIntDig

The long interdigit timer, in seconds. If you configure the “L” timer in the digit map, do not set this object.

pktcEUERSTNfBCallStatus

Set to **createAndGo(4)**. Setting only this object creates an entry with reasonable default values.

Configuring the Status Change Feature

In AR01.1, Status Change is used to periodically re-register with the network. One entry in the **pktcEUERSTUEActStatChgRegExp** is supported in AR01.1 and applies to the entire eDVA.

Follow these steps to configure the Status Change entry.

1. Set the **pktcEUERSTUEActStatChgRegExp** object to the desired registration expiration time, in seconds.
2. Set the **pktcEUERSTUEActStatChgStatus** object to **createAndGo(4)**.

Configuring No Answer Timeout

Follow these steps to configure the No Answer timeout feature. This feature allows the eDVA to disconnect a call if the called party does not answer before the timer expires.

1. Set the **pktcEUERSTNoAnsTODuration** object to the desired No Answer timeout, in seconds.
2. Set the **pktcEUERSTNoAnsTOSTatus** object to **createAndGo(4)**.

Configuring Caller ID

Configuring Caller ID requires setup of several tables. Follow these steps to configure Caller ID.

1. Set up the user's presentation by creating an entry in the **pktcEuerstCIDTable** with the following values:

pktcEuerstCIDPPS

Set the presentation status to **public**(2).

pktcEuerstCIDStatus

Set to **createAndGo**(4).

2. Set up the display feature by creating an entry in the **pktcEuerstCIDDisTable** with the following values:

pktcEuerstCIDDisDefCountry

Enter the country code data, to be stripped from the display information sent to the subscriber's CPE. For example, the US country code is **1**.

pktcEuerstCIDDisTimeAdj

The adjustment, in minutes, from UTC.

pktcEuerstCIDDisDSTFlag

Set to **1** to adjust for Daylight Savings Time, or **0** to ignore DST.

pktcEuerstCIDDisDSTInfo

To use the above two objects for adjusting the time, set this to an empty string. Otherwise, set to a POSIX timezone string as defined in RFC 4833. If you set this object, the eDVA ignores the above two objects.

pktcEuerstCIDDisCIDCWActStat

Set to **true**(1) to disable CID-CW.



Note: You can disable CID-CW for the entire eDVA by setting bit **0x02000000** of the SIP Feature Switch **sipCfgSipFeatureSwitch**.

pktcEuerstCIDDisStatus

Set to **createAndGo**(4).

Configuring Emergency Services

Follow these steps to configure emergency services. See [Emergency Calls](#) (page 81) for an overview of emergency call processing.

1. Add a rule to the digit map to associate a dial string with emergency calls. For example:
"911" : EMERGENCY-CALL ("sip: " "911" =domain =Emergencytg)
2. To configure PacketCable 2.0 emergency service features, provision the following objects in the **pktcEuerstNfEmSvcTable**. The index is *.oper.x*, where *oper* is the index of the entry in the Operator table.

pktcEuerstNfEmSvcNwHoldTimer

(non-TW loads only) The emergency services network hold timer value, in minutes.
Default: **45**.

pktcEUERSTNfEmSvcHowlTimer

The emergency services howler timer, in seconds. Default: **3**.

pktcEUERSTNfEmSvcDSCPValMedia

The DSCP value for network media (RTP) packets associated with emergency calls.

pktcEUERSTNfEmSvcDSCPValSig

The DSCP value for network signaling packets associated with emergency calls.

pktcEUERSTNfEmSvcStatus

Set to **createAndGo(4)** to write the entry and put it in service.

3. To change the URN used to identify incoming emergency calls, set the **sipCfgEmergencyServiceURN** object to the desired URN. The default setting is "URN: service: sos."
4. To disconnect active calls when originating or receiving an emergency call, set bit **0x00001000** in the **sipCfgSipFeatureSwitch** object.



Note: The eDVA does not notify the subscriber when disconnecting an active call in response to an incoming emergency call.

Configuring Distinctive Ring/Alert Tones

Setting Alert-Info strings for distinctive ringing and alert (Call Waiting) tones is identical for both PC20 and ARRIS SIP loads. See [Configuring Distinctive Ring/Alert Tones](#) (page 105) for details.

Configuring PacketCable 2.0 Digit Maps

The PacketCable *Residential SIP Telephony Feature Specification*, PKT-SP-RSTF-I08-110127, describes digit maps and provides an example map. The PacketCable 2.0 digit map specification is radically different from that defined in earlier specifications: earlier maps simply defined valid dialing sequences and left their interpretation to the Call Server (NCS) or vendor-defined methods (previous ARRIS SIP firmware versions use the **sipCfgDialFeat*** MIB objects); the new digit maps both define valid dialing sequences and associate actions with them.

General Digit Map Structure

A digit map consists of:

timer definitions

The following timers are defined by PacketCable 2.0 specifications. All are specified in seconds, and can be defined in 0.1 second increments.

- S (Short Interdigit Timer): when a subscriber's dialing string matches a pattern, but subsequent digits could match a longer pattern, pausing long enough for the "S" timer to expire selects the shorter pattern.

Default: 4 seconds.

- Z (Long Duration Timer): used in patterns to indicate that the following key is held down for a minimum amount of time.

Default: 2 seconds.

The following timers are supported in AR01.1 digit maps, but are deprecated. Use the appropriate MIB objects to set these timer values.

- T (Start Timer): the length of time allowed, after beginning to play dialtone, for the subscriber to dial the first digit. Typically, if this timer expires without the subscriber dialing anything, the UE plays reorder tone. Use the [pktcEUSERSTNfBCallOrigDTTimer](#) object to specify this timer.

Default: 16 seconds.

- L (Long Interdigit Timer): the allowable time between digits if the "S" timer is not specified. Use the [pktcEUSERSTNfBCallOrigModLongIntDig](#) object to specify this timer.

Default: 16 seconds.

symbol definitions

A [symbol](#) is a string constant, used to define common pattern or action strings and to make rules more human-readable. An example symbol definition is:

Local Number = [2-9]x{6}

map definitions

maps consist of a series of rules defined as:

pattern : action

Patterns are completely defined in the *Residential SIP Telephony Feature Specification*.

The following list describes the more common components of a pattern.

- Numbers 0 through 9, and the * and # symbols, represent standard keypad keys. The characters "x" and "X" represent a single numeric keypress.
- S, T, L, and Z represent the defined timers. Of these, the Z timer must precede a keypress.
- An = sign followed by a name represents a defined symbol. Example: =Local Number
- Square brackets enclosing a series of keys, such as [2345], match any single keypress in that series. A range of numeric keys may be represented using a hyphen, so [2-5] is equivalent to [2345]. The character "x" or "X" is equivalent to [0-9]. If the first character in the series is a caret (^), then the series matches a keypress not represented — for example, [^01*#] is equivalent to [2-9].
- A keypress (or a series enclosed in square brackets) followed by a number or range in curly brackets, such as x{6}, matches when the specified number (or some number within the range) of matching keys are pressed. For example, 1x{2-4}# matches 1xx#, 1xxx#, or 1xxxx#.

- All or part of a pattern may be enclosed in parentheses to specify a *sub-pattern*. A sub-pattern may be used in an action, as described below.
- Two slashes (//) indicate that the rest of the line is a comment.

Actions, like patterns, are completely defined in the *Residential SIP Telephony Feature Specification*. An action consists of a command, followed by a parameter string enclosed in parentheses. Multiple actions can be specified, and are separated by a semicolon (;). AR01.1 supports the following commands in actions:

- MAKE-CALL (*uri*) — URI representing the specified destination (usually a phone, but could include dialing codes handled in the network)
- CID-DELIVER (*string*) — send the specified Caller ID information
- CID-SUPPRESS (*string*) — do not send Caller ID information
- CW-TOGGLE — toggle Call Waiting
- EMERGENCY-CALL (*uri*) — call the specified emergency number
- HOLD-ACTIVATE — hold the active call
- COT-ACTIVATE — perform a Customer-Originated Trace of the last call
- RECALL — play Recall tone
- REORDER — play Reorder tone
- RETURN (*string*) — specifies the actual value returned by the map
- SB-MAINT (*string*) — solicitor blocking
- SD-PROGRAM (*vsc, string*) — adds a number to the Speed Dial list
- 3WC-ENABLE — enables 3-way calling (ARRIS extension).
- 3WC-DISABLE — disables 3-way calling (ARRIS extension).
- USEMAP (*=map-name*) — applies the specified map to input received after matching this rule
- CNDB-TOGGLE — toggles the Caller ID privacy setting for the next outgoing call.
- ACR-ACTIVATE — activates rejection of incoming calls with blocked Caller ID information.
- ACR-DEACTIVATE — allows incoming calls with blocked Caller ID information.
- FEATURE-CHECK (*featureID, [failureURI]*) — if the feature *featureID* is not enabled, the eDVA plays the specified *failureURI* (or reorder tone if not specified).
- DND-ACTIVATE — activates Do Not Disturb (ARRIS extension).
- DND-DEACTIVATE — deactivates Do Not Disturb (ARRIS extension).

The following is a subset of the more common components of parameters to action commands.

- A string enclosed in quotes, such as "*@tel co. example.net*", is a literal string.
- An = sign followed by a name represents a defined symbol. Example: *=Local Number*
- A pound sign (#) followed by a number represents a sub-pattern within a matched pattern. For example, if the defined pattern was (*[2-9]xx*) (*555(xxxx)*) and the dial string matched was *6785551212*, then *#2* returns the value "*5551212*." The pattern *#0* represents the entire matched dial string.
- A pound sign, followed by a number and a "v" specifies a sub-pattern returned by a map.

A series of components are concatenated; for example,

"sip: " #0 "@example.net" =dial string

constructs a single parameter string.

AR01.1 Compliance with PacketCable 2.0

AR01.1 generally conforms to PacketCable 2.0 specifications for digit maps, with the following exceptions:

- The Start Timer “T” is supported, with a default value of 16 seconds, and can be changed in the digit map. However, its behavior is hard-coded to play reorder tone, equivalent to defining the rule “T” : REORDER in the digit map. Configuring this timer in the digit map is deprecated; use the **pkteUERSTNfBCallOrigDTTimer** object to set this timer.
- The Long Interdigit Timer “L” can be defined but is not supported in rules. However, if the Short Interdigit Timer “S” is not defined, the “L” timer definition is applied to rules where “S” is specified. Configuring this timer in the digit map is deprecated; use the **pkteUERSTNfBCallOrigModLongIntDig** object to set this timer.
- The keys “A” through “D” are not supported.
- Symbols are supported, but the definition of a symbol must be a constant string. Thus, the following symbol definition is supported:

```
l o c a l N u m b e r = " [ 2 - 9 ] x { 6 } "
```

but this is not:

```
l d N u m b e r = " 1 " = A r e a C o d e = l o c a l N u m b e r
```

- The following action commands are not supported:
 - AC-ACTIVATE, AC-DEACTIVATE
 - AR-ACTIVATE, AR-DEACTIVATE
 - CFV-PROGRAM, CFV-DEACTIVATE
 - DA-MAINT
 - DND-PROGRAM
 - SCF-PROGRAM
 - SPP-PROGRAM

Specifying a Digit Map

Follow these steps to specify a digit map.

1. Use a text editor to create a digit map. Note the restrictions above.
2. In the configuration file, set the **pkteUERSTDMValue.1** MIB object to the text of the digit map. Note that you will need to use the **LongSnmpMib** TLV to accommodate the size of the map.



Note: In AR01.1, a single digit map applies to all lines in the E-UE.

Example Digit Map

The following is an example digit map.

```
// T i m e r v a l u e s
T I M E R T = 1 6 // S t a r t t i m e r . T h e l e n g t h o f t i m e a l l o w e d t o d i a l t h e
```



```

// first digit from the time dial tone is applied
TIMER S=4 // Short interdigit timer. Used when critical timing should
// be performed, such as when the dialed digits constitute
// a complete address, but additional digits may
// constitute a different complete address
TIMER L=16 // Long interdigit timer. The allowable time between digits
// if the short interdigit timer has not been indicated in
// the digit map.
TIMER Z=2.0 // Long duration timer. The duration a particular digit is
// to be held in order to be detected.
// Symbols
domain = "@tel.example.com"
areaCode = "303"
dialString = ";user=dialstring" // Just to shorten things
homeEmergencyNumber = "911"
localEmergencyNumber = "911" // alternate emergency number
lcltg = ";tgid=Local_trunk-group-id"
ldtg = ";tgid=Long_Distance_trunk-group-id"
intlgtg = ";tgid=International-trunk-group-id"
Emergencytg = ";tgid=911PSAP-trunk-group-id"
Opertg = ";tgid=Local_Operator-trunk-group-id"
ldcic = ";cic=0333"
intlclcic = ";cic=8937"
TollFreeclcic = ";cic=0110"
// Maps
MAP MainTable = // This is where processing starts
"T" : REORDER // Reorder Tone or Annc.
"OS" : MAKE-CALL ("sip:0" =domain =Opertg)
"0#" : MAKE-CALL ("sip:0" =domain =Opertg)
"00" : MAKE-CALL ("sip:0" =ldcic =domain =ldtg)
"(=Emergency)" : EMERGENCY-CALL ("sip:" "911" =domain =Emergencytg )
// map N11 to routing number or reorder if not assigned
"211" : MAKE-CALL ("sip:" "+13035551111" =domain =lcltg)
"311" : MAKE-CALL ("sip:" "+13035552222" =domain =lcltg)
"411" : MAKE-CALL ("sip:" "411" =domain =Opertg)
"511" : REORDER // Reorder Tone
"611" : MAKE-CALL ("sip:" "+13035552224" =domain =lcltg)
"711" : MAKE-CALL ("sip:" "+18885552225" =TollFreeclcic =domain =lcltg)
"811" : MAKE-CALL ("sip:" "+19035552226" =ldcic =domain =ldtg)
"(=Speedcall)" : MAKE-CALL ( "sip:" #1v)
"(=PhoneNumber)" : MAKE-CALL (#1v)
"(=ImmediateVSCs)" : RETURN
"(=DelayedVSCs)" : RETURN
// Press # for 2 seconds & get recall dial tone
"Z#" : RECALL; USEMAP(=MainTable)
// Any other digit strings
"(x{1-15})S" : REORDER
"(x{1-15})#" : REORDER
MAP PhoneNumber =
"(=LocalPhoneNumbers)" : RETURN ("+" #1v =domain =lcltg)
"0(=LocalPhoneNumbers)" : RETURN ("0" #1v =domain =Opertg)
"1{0-1}([579]00[2-9]x{6})" : RETURN ("+" #1 =domain =lcltg)
// 500, 700, & 900 numbers
"1{0-1} (=TollFreeNumbers)" : RETURN ("+" #1v =TollFreeclcic =domain
=lcltg)
"101(=DialAround)" : RETURN ( "sip:" #1v ";dai=no-presub" =domain =lcltg)
"1(=WZ1InternationalNumbers)" : RETURN ("+" #1v =intlclcic =domain =intlgtg)
"0(=WZ1InternationalNumbers)" : RETURN ("0" #1v =intlclcic =domain =intlgtg)
) "011(=InternationalNumbers)" : RETURN ("+" #1v =intlclcic =domain =intlgtg)
"01(=InternationalNumbers)" : RETURN ("01" #1v =intlclcic =domain =intlgtg)
"1{0-1} (=LDPhoneNumbers)" : RETURN ("+" #1v =ldcic =domain =ldtg)
"0(=LDPhoneNumbers)" : RETURN ("0" #1v =ldcic =domain =ldtg)
MAP Emergency = // Matches emergency dial
"(=localEmergencyNumber)" : RETURN
"(=homeEmergencyNumber)" : RETURN

```



```

"[01](=homeEmergencyNumber)" : RETURN
MAP Speedcall = // Matches Speed Call list (either one or two digit)
// two-digit speed dialing
"21S" : RETURN ( "sip:" "+19137654321" =ldcic =domain =ldtg)
"22S" : RETURN ( "sip:" "+13037654321" =domain =lcltg)
"23S" : RETURN ( "sip:" "+18007654321" =TollFreeCic =domain =lcltg)
"24S" : RETURN ( "sip:" "+529137654321" =intlCic =domain =intlTG)
"21#" : RETURN ( "sip:" "+19137654321" =ldcic =domain =ldtg)
"22#" : RETURN ( "sip:" "+13037654321" =domain =lcltg)
"23#" : RETURN ( "sip:" "+18007654321" =TollFreeCic =domain =lcltg)
"24#" : RETURN ( "sip:" "+529137654321" =intlCic =domain =intlTG)
MAP LocalPhoneNumbers = // Matches local phone numbers
"(=areaCode)(=Local7DigitNumbers)" : RETURN( =areaCode #2v )
"( =Local7DigitNumbers )" : RETURN( =areaCode #1v )
MAP Local7DigitNumbers = // Matches local 7 digit numbers
"(212x{4})" : RETURN( #1 )
"(213x{4})" : RETURN( #1 )
"(222[1-3]x{3})" : RETURN( #1 )
"(2267x{3})" : RETURN( #1 )
// could be 100+ entries for local digit map
MAP WZ1InternationalNumbers = // Matches international WZ1 numbers
"(204x{7})" : RETURN( #1 )
// 42 more entries for area codes of Canada & Caribbean
MAP TollFreeNumbers = // 800 and friends
"(800[2-9]x{6})" : RETURN( #1 )
"(866[2-9]x{6})" : RETURN( #1 )
"(877[2-9]x{6})" : RETURN( #1 )
"(888[2-9]x{6})" : RETURN( #1 )
MAP LDPhoneNumbers = // Matches Long Distance phone#
// Local, 500, 700, 800, 900 & WZ1 International
// eliminated by prior matches
"([2-9]x{6})S" : RETURN( =areaCode #1 )
"([2-9]x{6})#" : RETURN( =areaCode #1 )
"([2-9]x{9})" : RETURN( #1 )
MAP InternationalNumbers = // Matches international non-WZ1 numbers
"([2-9]x{1-14})S" : RETURN( #1 )
"([2-9]x{1-14})#" : RETURN( #1 )
MAP DialAround = // Matches dial around phone#
"0([2-9]x{2})(=daiPhoneNumbers)" : RETURN( #2v ";cic=0" #1 )
// (3 digit CIC dialed by user)
"1(x{4})(=daiPhoneNumbers)" : RETURN( #2v ";cic=1" #1 )
// (4 digit CIC dialed by user)
MAP daiPhoneNumbers = // valid dai phone numbers
"[5789]00" : REORDER
"S" : RETURN
"# " : RETURN
"866" : REORDER
"877" : REORDER
"888" : REORDER
"(=LocalPhoneNumbers)" : RETURN (" +1" #1)
"0(=LocalPhoneNumbers)" : RETURN ("0" #1)
"1{0-1} (=WZ1InternationalNumbers)" : RETURN (" +1" #1)
"0(=WZ1InternationalNumbers)" : RETURN ("0" #1)
"011(=InternationalNumbers)" : RETURN ("+" #1)
"01(=InternationalNumbers)" : RETURN ("01" #1)
"1{0-1}(=LDPhoneNumbers)" : RETURN (" +1" #1)
"0(=LDPhoneNumbers)" : RETURN ("0" #1)
MAP ImmediateVSCs = // Matches and executes immediate VSCs.
// Returns nothing.
"*74" // (SD8)
: RECALL; USEMAP (=SD8)
"*75" // (SD30)
: RECALL; USEMAP (=SD30)
"*[78]7" // (ACR- ACTI VATE/ACR- DEACTI VATE)

```

```

: MAKE-CALL ( "sip: " #0 =domain =dialString)
"*9[01]" // (DND- ACTI VATE/DND- DEACTI VATE)
: MAKE-CALL ( "sip: " #0 =domain =dialString)
"*96" // (SB- MAINT)
: SB- MAINT (#0) // solicitor blocking maintenance
"*72" // (CFV- ACTI VATE)
// Play Recall Dial Tone
// Collect digits using subset digitmap
// Prepend "*72." onto dialed address using SIP URI
// The following is an example implementation:
: RECALL; USEMAP(=ForwardingNumber)
"*73" // (CFV- DEACTI VATE) (reuse *72 for this)
: MAKE-CALL ( "sip: *72." =domain =dialString)
MAP DelayedVSCs = // Make some state change, then continue processing
dialing
"*92" : HOLD- ACTI VATE; RECALL; USEMAP(=Mai nTabl e)
"*67" : CID- SUPPRESS
"*82" : CID- DELIVER
"*70" : CW- TOGGLE; RECALL; USEMAP(=Mai nTabl e) // The CALL- WAITING action
must give the right tone.
MAP SD8 = // Program one- digit speed dial number
"[2-9]" : SD- PROGRAM ( "sip: *74." #0 =domain =dialString)
"[^2-9]" : REORDER
"S" : REORDER
"# " : REORDER
MAP SD30 = // Program two- digit speed dial number
"[2-4]x" : SD- PROGRAM ( "sip: *75." #0 =domain =dialString)
"[^2-4]" : REORDER
"S" : REORDER
"# " : REORDER
"[2-4]S" : REORDER
MAP ForwardingNumber = // Just for programming CFV
"(=PhoneNumber)" : MAKE- CALL ( "sip: *72." #1 =domain =dialString)

```

Provisioning PacketCable 2.0 Users

Use this procedure to set up users and associate them with credentials and lines.

Action

Follow these steps:

1. If using Certificate Bootstrapping, skip to step 3.
2. Configure an IMPI table entry in the eDVA configuration file. The index is usually 1. Set the objects in this entry as follows:

pktcEUEUsrIMPIldType

The identification type for the private identity. Only **privateIdentity(4)** is supported in AR01.1 and is the default.

pktcEUEUsrIMPIld

The user's login information used to authenticate the eDVA to the headend.

pktcEUEUsrIMPICredsType

The type of credentials used to confirm the private identity. Only **password(3)** is supported in AR01.1 and is the default.

pkcEUEUsrIMPICredentials

The user's password. Reading this object always returns an empty string.

pkcEUEUsrIMPIRowStatus

Set to **createAndGo**(4).



Note: In most cases, a single IMPI table entry is associated with all public users.

3. Configure an IMPU table entry in the eDVA configuration file. Set the objects in this entry as follows:

pkcEUEUsrIMPUIType

(optional) The public user identification type. Both **publ i c I d e n t i t y**(3) (the default) and **user Name**(6) are supported in AR01.1.

pkcEUEUsrIMPUIId

The public user identification. The content of this object depends on the value of

pkcEUEUsrIMPUIType:

- **publ i c I d e n t i t y**(3): *phonenumber@domain*
- **user Name**(6): *phonenumber* (Touchstone firmware adds the default domain to the phone number to create the full public identity)

pkcEUEUsrIMPUIMPIIndexRef

The index into the **pkcEUEUsrIMPITable**, specifying the IMPI entry associated with the public user.

pkcEUEUsrIMPUDispInfo

The display name in SIP messages, associated with Caller ID name delivery.

pkcEUEUsrIMPUPopIndexRefs

The index into the **pkcEUEDevOpTable**, specifying the operator entry associated with the public user. In AR01.1, this is usually **1** since only one operator entry is supported.

pkcEUEUsrIMPUPAdminStat

(optional) Accept the default of **act i v e**(1) to enable the line.

pkcEUEUsrIMPUPAdditionalInfo

If only one public user is associated with a phone line, the eDVA automatically maps the first user to line 1 and the second user to line 2. For multiple phone numbers associated with a line, as is the case with a Teen Line service or an NPA or exchange transition, set this object as follows:

IEP# line;OEP# line

The *line* is the **ifIndex** for the line (usually **9** for line 1 and so on), and must be same for both IEP (Input EndPoint) and OEP (Output EndPoint).

pkcEUEUsrIMPUPRowStatus

Set to **createAndGo**(4).

4. (TS7.6 MSUP2 and newer) Configure the optional Certificate Bootstrapping feature, by modifying the configuration file, as follows:
 - a. Verify that the CableLabs Service Provider Root certificate in the E-UE and the Certificate Bootstrapping server are identical.
 - b. Set the **pktcEUECBEnable** object to **true**(1).
 - c. Set the **pktcEUECBData** object to the HTTPS URL of the Certificate Bootstrapping configuration file. This is an XML file, and is downloaded using HTTP over TLS.
 - d. Reset the eDVA to use the new provisioning.
 - e. (optional) To verify operation, walk the **pktcEUEUsrIMPITable** to see the settings.

Provisioning PacketCable 2.0 Application Profiles

Use this procedure to provision the Application Profile. The profile collects feature sets to associate with a user.

Before you can provision the Application Profile, you must configure the features as described in [Provisioning PacketCable 2.0 Features](#) (page 118).

Indexing

The **pktcEUERSTAppProfileToFeatTable** uses a dual index for each entry: *.profile.feature*.

Action

Follow these steps to provision an Application Profile. The profile must be provisioned in the eDVA configuration file.

1. Set the **pktcEUERSTAppFeatID** to the feature type being specified. See [Provisioning PacketCable 2.0 Features](#) (page 118) for a list of valid feature types.
2. Set the **pktcEUERSTAppFeatIndexRef** object to the index of an entry in the **pktcEUERSTAppProfileToFeatTable**. This specifies the feature definition to use.
3. (optional) Set the **pktcEUERSTAppAdminStat** object to **active** (the default) to activate this feature, or **inactive** to disable this feature.
4. Set the **pktcEUERSTAppStatus** to **createAndGo**(4).

Provisioning PacketCable 2.0 Application Maps

Use this procedure to associate an application and an application profile with a user.

Prerequisites

Before provisioning an application map, perform the following procedures:

1. [Provisioning PacketCable 2.0 Features](#) (page 118)
2. [Provisioning PacketCable 2.0 Application Profiles](#) (page 132)
3. [Provisioning PacketCable 2.0 Users](#) (page 130)

Indexing

The **pktcEUEUsrAppMapTable** uses a dual index for each entry: *.user.map*, where *user* index is the index of the IMPU entry that this map is associated with; the *map* is the application index (usually **1** since AR01.1 supports only the PacketCable RST application).

Action

Follow these steps to provision an application map. You must add these objects to the eDVA configuration file.

1. Set the **pktcEUEUsrAppMapAppOrgID** object to the enterprise number of the organization defining the application: always **4491** for PacketCable.
2. Set the **pktcEUEUsrAppMapAppIdentifier** object to the application: always **1** for Residential SIP Telephony (RST).
3. Set the **pktcEUEUsrAppMapAppIndexRef** object to the index of the entry in the **pktcEUEUsrAppMapAppProfileToFeatTable** corresponding to the profile associated with this map.
4. (optional) Set the **pktcEUEUsrAppMapAppAdminStat** object to **active** (default) to enable the application for this user, or **inactive** to disable the application.
5. Set the **pktcEUEUsrAppMapRowStatus** object to **createAndGo(4)**.

Configuring SIP Failure Response Tones

Touchstone firmware complies with PacketCable 2.0 specifications for playing tones in response to various SIP failure messages. Use this procedure to override the default behavior.

Priority

The eDVA follows these steps to determine which tone to play when receiving a failure response.

1. If the ([sipCfgSipFeatureSwitch](#)) object has the **playBusyToneOnReject**(0x00000400) bit set, the eDVA always plays busy tone.
2. If the **playBusyToneOnReject**(0x00000400) bit is not set, and the P-CSCF specifies a tone in the Error-Info header of the failure response, the eDVA plays the specified tone.
3. If the [pkcEuerSTNfAncMapURL.code](#) object corresponding to the response code is set, the eDVA plays the configured tone.
4. The eDVA plays the default response tone as defined in the table below:

Response Code	Tone
486	Busy
487	Silence
600	Busy
Other (except 401 or 407)	Reorder

Action

Perform the following tasks as needed.

- [Playing Busy Tone for All Errors..... 134](#)
- [Configuring Individual Response Tones..... 135](#)

Playing Busy Tone for All Errors

To override PacketCable 2.0 behavior and play busy tone for all failure responses, follow these steps.

1. Set the **0x00000400** bit of the [sipCfgSipFeatureSwitch](#) in the eDVA configuration file.
2. Restart the eDVA to make the change take effect.

Configuring Individual Response Tones

Follow these steps to configure a tone to play for a specific failure response.

1. For each response code, set the **pktcEuerSTNfAncMapRspCode** object, either in an SNMP manager or in the eDVA configuration file. This object must be indexed by both the domain entry (AR01.1 allows only one domain per E-UE) and the response code; for example, **pktcEuerSTNfAncMapRspCode.1.404**, and contain the URI of the PacketCable-defined tone to play. For example:

```
{pktcEuerSTNfAncMapRspCode.1.404 "file:///PacketCableRST/bz"}
```

The following failure codes are supported:

- 404, 406, 408
- 480, 484, 486, 487
- 500, 503, 504
- 600, 603



Note: A URI sent in the Error-Info header of the failure response takes precedence over these settings.

2. If using an SNMP manager, set the **pktcEuerSTNfAncMapStatus** object, indexed by the same domain and response code, to **createAndGo(4)**. Otherwise, restart the eDVA to make the change take effect.
3. To remove an entry, set the **pktcEuerSTNfAncMapStatus** object, indexed by the same domain and response code, to **destroy(6)**.



Note: Attempting to remove the entry for response codes 486, 487, or 600 resets the entry to its default value.

Configuring MWI Support

Touchstone firmware provides improved control over both visual and audible MWI (Message Waiting Indication) indicators.



Note: The tasks in this procedure apply only to PacketCable 2.0 SIP deployments. ARRIS-proprietary legacy SIP deployments are not supported.

Action

Perform the following tasks as needed.

- [Clearing MWI Indicators](#) 136
- [Provisioning the MWI Subscription](#) 136
- [Voice Mail Subscription Watchdog](#) 136

Clearing MWI Indicators

It may be necessary to clear MWI indicators in response to a subscriber trouble call. Follow these steps to clear the indicators.

1. Set the **sipCfgPortMWIClear** line object to **true**(1). The line is the line number for which the indicators should be cleared; use **1** for line 1, and so on.
2. Repeat step 1 for each line that needs to be cleared.

Provisioning the MWI Subscription

By default, the eDVA subscribes to the P-CSCF MWI package (message-summary). Follow these steps to provision the eDVA to subscribe to a different server's MWI package.

1. Set the **sipCfgMWITargetAddrType** object to **ipv4**(1) or **ipv6**(2).
2. Set the **sipCfgMWITargetAddr** object to the IP address of the device that the eDVA should subscribe the MWI package to.
3. (optional) Set the **sipCfgMWITargetPort** object to the port number of the MWI service. The default is **5060**.
4. To recommend a subscription duration, set the **pktcEUERSTNfMWISubDuration** object to the recommended time before the subscription expires.

Valid range: **0** to **4000000** seconds. Default: **3600**.

Voice Mail Subscription Watchdog

In this AR01.1 release, PC20 SIP loads support automatic re-subscription to MWI. The new **sipCfgMWISubscriptionCheckInterval** object specifies the interval, in minutes, that the eDVA checks the MWI subscription state. At each interval, if the line is operational and MWI is unsubscribed, it automatically restarts the subscription.

The default value of 0 disables this feature.

Provisioning a Gateway (eRouter)

Use this procedure to provision the eRouter module in a Touchstone Gateway product. The eRouter provides up to four Ethernet connections and 802.11b/g/n/ac wireless (wifi) service. The wireless interface supports up to eight SSIDs, two of which are provisionable by the subscriber.

Touchstone Gateway products conform to the eRouter specification, CM-SP-eRouter-I06-110623.

For detailed business services configuration instructions, see the *Business Service Configuration Feature Sheet* for your Touchstone Gateway model.

Gateway Provisioning Methods

You can provision the router using any of the following methods:

- eRouter (TLV-202) sub-TLVs in the CM configuration file, as defined in Section B.4 of CM-SP-eRouter
- TR-069 using TR-181v2 objects (use TR69ManagementServer/TLV-202.2 parameters in the CM configuration file to specify ACS connection parameters)
- SNMP (using the CLAB-WIFI-MIB and a subset of the RDKB-RG-MIB)
- web pages (webGUI) — for details, see the *ARRIS Router Setup Web GUI User Guide*.

There is no order of preference among the provisioning methods; the last change before a commit is the change applied.

Provisioning Precedence

For AR01.1, subscriber configuration changes have the highest precedence, and override post-provisioning configuration.

Any settings the subscriber leaves at default may be changed. In this case, the last setting made, by any of the following methods, is the setting used:

- SNMP management
- Web-based interface
- TR-069
- Technician CLI

Changes made by the above methods are persistent, and are applied after the router processes changes made by DHCP and provisioning files.

eRouter Operating Modes

The eRouter operates in one of the following modes.

Router (NAT mode)

This is the default mode. The router provides NAT connectivity to devices connecting on the LAN.

In Router mode, the router is treated as a CPE device. This requires setting the **MaxCPE** TLV to **2** or higher.

Bridge (AP mode)

In Bridge mode, the router simply forwards packets between the LAN and WAN interfaces. The wireless interface and web interface function as normal.

In Bridge mode, the router is not treated as a CPE device.

Any of the operating modes can be selected through provisioning.

eRouter Wi-Fi Country Codes

To enable support for a specific country, set one of the following:

- TR-069: Device.WiFi.Radio.10{i}00.RegulatoryDomain (where {i} is 0 for the 2.4 GHz radio or 1 for the 5.0 GHz radio)
- SNMP: **clabWiFiRadioRegulatoryDomain** or **rdkbrgDot11ExtCountry** objects (in the CM configuration file)

The following values are supported:

- **worldwide**(0)
- **thailand**(1)
- **israel**(2)
- **jordan**(3)
- **china**(4)
- **japan**(5)
- **usa**(6)
- **europa**(7)
- **allchannels**(8)

The **worldwide**(0) setting is legal for all supported countries, but provides a minimal set of supported channels and power levels. The **allchannels**(8) setting should be used only for testing purposes.

LAN-side Devices

LAN-side devices are:

- Ethernet ports
- WiFi Basic Service Sets
- MoCA
- Wireless extenders

Internally, these devices are grouped together into routed network devices, described next.

Routed Network Devices

Routed network devices are groups of one or more LAN-side devices, presented as a single device. The following table describes the default mapping.

Device	Mapped interfaces	Subnet
1	SSID1, Ethernet 1-4, MoCA, USB	192.168.0.x
2	SSID2	192.168.26.x
3	SSID3	192.168.27.x
4	SSID4	192.168.28.x
5	SSID5	192.168.29.x
6	SSID6	192.168.30.x
7	SSID7	192.168.31.x
8	SSID8	192.168.32.x

Default eRouter Settings

The following table shows default eRouter settings.

Setting	Value
Operating Mode	Router
Default gateway IP address	192.168.0.1

MoCA Configuration Notes

DOCSIS 3.1 Gateway devices support MoCA 1.1 and MoCA 2.0 operation. MoCA (Multimedia over Coax Alliance) provides a simple way to extend network connectivity at the subscriber premises, using existing in-home coax cabling.

MoCA operation requires a PoE filter to be installed at the subscriber's network interface, before any splitters. Additional PoE filters may be required to block MoCA signals at older analog televisions or set-top boxes.

MoCA configuration is minimal; in most cases, you can accept the default provisioning. The TR-069 Device.MoCA parameters, and the MoCA-MIB, contain provisioning and administrative objects.

eRouter IPv6 Operation

AR01.1 supports IPv6 on the eRouter. The default eRouter configuration automatically provides *dual-stack* operation, assigning both IPv4 and IPv6 addresses to CPE devices connecting through Ethernet or Wifi.

The gateway can assign IPv6 addresses to CPE devices using:

- DHCPv6
- Stateless Address Auto-Configuration (SLAAC)

Requirements

When provisioned for IPv6, the Gateway requires an IPv6 address with a 56-bit prefix, either from DHCPv6 or by static assignment. This is required for proper CPE device connection.

When provisioning client filters, the end address must be higher than the start address. If the end address is lower than the start address, client filters fail without warning.

See "Configuring IPv6 CPE addressing" below for configuration details.

TR-069 Provisioning

Touchstone firmware supports basic gateway provisioning through TR-069. Other provisioning methods — including SNMP and the web-based interface — are also supported.

Enabling TR-069 Support

To enable TR-069 support, set the `rdkbTR069ClientMode.0` object to `true(1)` in the CM configuration file. The default is `false(2)`.

AR01.1 uses TLV-43 based provisioning, described below.

TR-069 may be automatically disabled in the following ways:

- The ACS sets the parameter `Device.ManagementServer.EnableCWMP` to false.
- The DHCP server does not send an ACS URL in DHCP option 125.

See [TR-069 Management](#) (page 295) for available objects and their descriptions.

TR69AcInfo Sub-TLV Formal Definitions

The following define the sub-TLVs in the **TR69AcInfo** group.

Tr69AcsUrl

The string defines the URL of the ACS server.

Type	Length	Value
43.9.1	variable	string

TR69AcsUserName

The string defines the user ID for logging into the ACS server.

Type	Length	Value
43.9.2	variable	string

TR69AcsPassword

The string defines the password for logging into the ACS server.

Type	Length	Value
43.9.3	variable	string

TR69CrUserName

The string defines the user name for the ACS when using the Connection Request mechanism to the Gateway.

Type	Length	Value
43.9.4	variable	string

TR69CrPassword

The string defines the password for the ACS when using the Connection Request mechanism to the Gateway.

Type	Length	Value
43.9.5	variable	string

TR69AcPerInfEnable

Enables or disables Periodic Inform. When enabled, the CPE sends periodic status information to the ACS using an inform method.

Type	Length	Value
43.9.6	1	boolean

TR69AcPerInfInterval

The time, in seconds, that the CPE must attempt to connect to the ACS to send a Periodic Inform.

Type	Length	Value
43.9.7	1	integer

TR69AcXAllowDocsCfg

Enables or disables storage of ACS parameters to non-volatile memory.

Type	Length	Value
43.9.8	1	boolean

TR69CrURL

The URL to the Gateway that the ACS uses for a Connection Request. The URL must be specified as: `http://gatewayIP:15627/acscal`

Type	Length	Value
43.9.9	variable	string

Obtaining TR-181 Parameters Using DHCP

This <TouchstoneFWrelease> release supports the acquisition of certain TR-181 data objects using the Gateway DHCP sequence. These parameters are necessary for configuration of ACS contact information.

Touchstone firmware supports TR-069 discovery from the DHCP server by including the string `dsl forum.org` in one of the following DHCP options:

- DHCPv4 Option 124 (Vendor-Identifying Vendor Class)
- DHCPv4 Option 60 (Vendor Class Identifier)
- DHCPv6 Option 16 (Vendor Class)

The following TR-181 parameters may be obtained using DHCP:

Description	Option Number			TR-181 Parameter
	Option 43	Option 125	Option 17 (DHCPv6)	
ACS URL	1	11	1	ManagementServer.URL
Provisioning code	2	12	2	DeviceInfo.ProvisioningCode
CWMP Retry Minimum Wait Interval	3	13	3	ManagementServer.CWMPRetryMinimumWaitInterval
CWMP Retry Interval Multiplier	4	14	4	ManagementServer.CWMPRetryIntervalMultiplier

TLV-202 Based Provisioning

AR01.1 supports provisioning of ACS connection parameters, using the **eRouter** TLV (TLV-202) in the CM configuration file.

Touchstone firmware supports objects in sub-TLV 2 (**TR69ManagementServer**).

Example Configuration

The following is an example configuration file segment.

```
eRouter =
  TR69ManagementServer =
    EnableCWMP = 1
    URL = "https://acs.example.com:8048/service/cwmp"
    Username = "testUname"
    Password = "testPwd"
    ConnectionRequestUsername = "testCRUname"
    ConnectionRequestPassword = "testCRPwd"
    ACSOverride = 1
```

TR69ManagementServer Sub-TLV Definitions

The following define the sub-TLVs in the **TR69ManagementServer** group.

EnableCWMP

Enables or disables CWMP (CPE WAN Management Protocol). If disabled (the default), Touchstone firmware does not send Inform messages to the ACS, or accept Connection Request notifications from the ACS.

Type	Length	Value
202.2.1	1	0 or 1

URL

Specifies the URL of the ACS.

Type	Length	Value
202.2.2	variable	string

Username

The user name used to authenticate the CPE when connecting the the ACS using CWMP.

Type	Length	Value
202.2.3	variable	string

Password

The password used to authenticate the CPE when connecting to the ACS using CWMP.

Type	Length	Value
202.2.4	variable	string

ConnectionRequestUsername

The username used to authenticate an ACS making a Connection Request to the CPE.

Type	Length	Value
202.2.5	variable	string

ConnectionRequestPassword

The password used to authenticate an ACS making a Connection Request to the CPE.

Type	Length	Value
202.2.6	variable	string

ACSOVERRIDE

Supported but not used.

If enabled, the Touchstone device accepts the ACS URL from the CM configuration file, even if the ACS has overwritten the URL.

Type	Length	Value
202.2.7	1	0 or 1

Gateway DHCP Interactions

The eRouter portion of Touchstone Gateways interacts with the provisioning server, using the DHCP options described below.

WAN Interface Dynamic Provisioning

In IPv4 mode, the gateway requests the following provisioning, using Option 55 in a DHCP Discover or Request message:

- Option 1 (Subnet mask)
- Option 3 (Router)
- Option 15 (Domain name)
- Option 51 (IP address lease time)
- Option 54 (Server identifier)
- Option 55 (Parameter request) with the following sub-options:
 - Sub-option 1 (subnet mask)
 - Sub-option 3 (router)
 - Sub-option 6 (domain name server)
 - Sub-option 51 (IP address lease time)
 - Sub-option 54 (server identifier)
- Option 100 (Time zone)

IPv6 eRouter Considerations

When provisioned for IPv6 addressing, the eRouter expects an IPv6 address with a /64 prefix, either from DHCPv6 or by static assignment. This is a requirement for CPE devices to successfully connect to the eRouter. Optionally, the eRouter may receive a /56 Prefix Delegation subnet, for use as needed on the LAN side; each SSID would receive a /64 portion of the Prefix Delegation subnet.

Setting the Operating Mode

Follow these steps to configure the router operating mode. By default, the eRouter is enabled and supports dual-stack (IPv4 and IPv6) operation.

1. To change the operating mode, use one of the following methods:
 - Set TLV-202.1 to one of the following values:
 - 0 (bridge mode, router disabled)
 - 1 (IPv4 only)
 - 2 (IPv6 only)
 - 3 (IPv4 and IPv6) (default)
 - Set the **rdkbRgDeviceMode** object to one of the following values:
 - **multiSsid**(1) (legacy IPv4 router mode)
 - **ipv4**(3) (IPv4 only)
 - **ipv6**(4) (IPv6 only)
 - **dual stack**(5) (IPv4 and IPv6)

Note: Setting this object with an SNMP browser reboots the eRouter.
 - Set the **rdkbRgIpMgmtLanMode** object to one of the following values:
 - **bridge**(1) (bridge mode, router disabled)
 - **router**(2) (router mode, uses **rdkbRgDeviceMode** to further define router operation)
 - **mixed**(4) (mixed mode, uses **rdkbRgIpMgmtPortControlTable** to specify router operation per port)
 - Set the **esafeRouterInitModeControl** object to one of the following values:
 - **ipDisabled**(1) (bridge mode, router disabled)
 - **ipv4only**(2) (IPv4 only)
 - **ipv6only**(3) (IPv6 only)
 - **ipv4Andipv6**(4) (IPv4 and IPv6)
 - **honorRouterInitMode**(5) (honor TLV-202.1 settings)

TLV-202.1 overrides the SNMP settings.

2. If you make this change in the configuration file, restart the modem.

Configuring the Wireless Channel

The default eRouter configuration auto-selects a wireless channel, and is recommended for most environments.

1. To set channel auto-detect, use one of the following:
 - Using TR-069, set the Device.WiFi.Radio.{i}.AutoChannelEnable parameter to 1 for each radio.
 - Using the CableLabs WIFI-MIB, set the **clabWIFIRadioAutoChannelEnable** object to **true(1)** for each radio.
 - Using the RDKB MIB, set the **rdkbRgDot11ExtCurrentChannel.32** and **rdkbRgDot11ExtCurrentChannel.112** objects to **0**.
2. To set a particular channel use one of the following:
 - Using TR-069, set the Device.WiFi.Radio.{i}.Channel parameter to the desired channel for each radio.
 - Using the CableLabs WIFI-MIB, set the **clabWIFIRadioChannel** object to the desired channel for each radio.
 - Using the RDKB MIB, set the **rdkbRgDot11ExtCurrentChannel.32** and **rdkbRgDot11ExtCurrentChannel.112** objects to the desired channels.

The valid range depends on the country code. See WiFi Country Codes for a complete list of supported country codes.

Configuring DNS Override/Relay

The gateway supports DNS relay through TR-069 provisioning. Follow these steps to configure DNS Relay from SNMP.

1. Set Device.DNS.Relay.Enable to True to enable DNS Relay. (This value is True by default.)
2. For each external DNS server, set up a row in the Device.DNS.Relay.Forwarding.{i} table. Set the following parameters in the row:

Enable

Set to True to enable this row.

DNSServer

The IP of the DNS server to receive forward queries.

Interface

A path name to a row in the Device.IP.Interface table, specifying the interface used to forward DNS queries. Specify an empty string to use the default device routing policy.

Configuring IPv6 DHCP Services

Follow these steps to configure the eRouter DHCPv6 server, using TR-069.

1. Set Device.DHCPv6.Server.Enable to True.
2. For each logical LAN subnet (200 to 207), configure the corresponding entry in the Device.DHCPv6.Server.Pool.{i} table:
 - Enable: set to True.
 - Interface: a path name to a row in the Device.IP.Interface table, specifying the interface this pool entry serves.
3. (optional) For each logical LAN subnet (200 to 207), configure rows in the Device.DHCPv6.Server.Pool.{i}.Option.{i} table, to specify options offered to clients:
 - Enable: set to True.
 - Tag: the DHCPv6 Option code.
 - Value: the hexbinary-encoded option value.

Managing Network Extenders (AR01.1)

Use the Gateway web pages to manage network extenders.

Touchstone Gateways auto-discover network extenders that support the Home Network Extender (HNE) protocol. If the ARRIS Home Network Controller (AHNC) is active, the Gateway can display a topology of the subscriber's network, and push firmware updates to the network extenders.

Managing Network Extenders from the Gateway

Subscribers can access the Gateway pages from their LAN devices. If Remote Access is enabled, a network operator can access these pages from the NOC. To set up Remote Access, see the *Touchstone Router Web GUI User Guide*.

1. After logging into the router, use the menu on the left to navigate to Connected Devices -> Range Extenders.

From this screen, you can view connection details for each network extender associated with the subscriber's network.
2. To enable firmware upgrades for network extenders, use the menu of the left to navigate to AHNC -> Config.

Enable AHNC, if necessary, then enable HNE Firmware Upgrade.

Managing Network Extenders Using SNMP

1. To see all information about connected network extenders, walk the **rdkbRgNetworkExtenders** MIB. This displays two sets of tables:
 - **rdkbRgNetworkExtenderTable**—displays name, model numbers, and IP addresses of all connected network extenders.
 - **rdkbRgNetworkExtenderTable**—for each network extender, displays radio configurations including SSID, BSSID, and channel.

Note: All information in the **rdkbRgNetworkExtenders** MIB is read-only.

Provisioning Home Hotspot

AR01.1 loads support Home Hotspot functionality. This section describes details and provisioning information.

Supported Hardware

Home Hotspot is supported on the following Gateway models:

- DG34xx (all models)
- TG34xx (all models)

Home Hotspot Functionality

Touchstone devices supporting Home Hotspot act as the Home Hotspot Gateway (HHG) in the HFC network. The HHG supports two types of connections:

- Bridged: Traffic from a designated SSID on the Touchstone device is bridged to the Internet. Connected devices count against the configured MaxCPE limit unless **rdkbRgWifiHotspotIgnoreMaxCpeSetting** is set to **true**(1). Authentication is handled on the Gateway.
- GRE: Hotspot traffic uses a GRE tunnel from a designated SSID on the Touchstone device, to a WLAN gateway beyond the CMTS. A connected RADIUS server handles authentication. AR01.1 supports one GRE tunnel.

Management

The **rdkbRgWifiHotspotConnectedClientsTable** provides a list of associated devices using the Hotspot SSID.

The HHG uses DHCP Relay to forward DHCP traffic between connecting devices and the WLAN gateway. If necessary, provisioning can insert an optional clientID and remoteID in outgoing DHCP Discover, Request, and Inform messages.

Two MIB objects provide the IP address and type of the Active Tunnel Endpoint:

rdkbRgL2ogrePriRemoteAddress

The IP address of the Active Tunnel Endpoint.

rdkbRgL2ogrePriRemoteAddressType

The IP address type (IPv4 or IPv6) of the Active Tunnel Endpoint.

Action

Follow these steps to configure the Hotspot. These are the minimal steps required to configure the Hotspot; you can set other objects as needed to configure optional functionality or adjust operation as needed.

1. Set the following objects in the **rdkbRgWifiHotspotTable**. The index for this table identifies the SSID to use; 1 to 8 for the 2.4 GHz LAN and 9 to 16 for the 5 GHz LAN.

rdkbRgWifiHotspotMode

Set to **disable**(1) to disable Hotspot on this SSID. Set to **enableBridge**(2) for bridge mode, and **enableL2ogre**(3) for GRE tunnel mode.

rdkbRgWifiHotspotRadiusAccAddressType

The address type (IPv4 or IPv6) of the RADIUS accounting server.

rdkbRgWifiHotspotRadiusAccAddress

The IP address of the RADIUS accounting server.

rdkbRgWifiHotspotRadiusAccKey

The RADIUS accounting server shared security key.

2. If using bridge mode, set **rdkbRgWifiHotspotIgnoreMaxCpeSetting** to **true**(1) so Hotspot clients do not count against the configured MaxCPE setting.
3. If using GRE tunnel mode, set the following **rdkbRgL2ogreBase** objects:

rdkbRgL2ogreEnabled

Set to **true**(1) to enable GRE tunnel support.

rdkbRgL2ogrePriRemoteAddressType

The address type (IPv4 or IPv6) for the primary remote endpoint of the GRE tunnel.

rdkbRgL2ogrePriRemoteAddress

The IP address for the primary remote endpoint of the GRE tunnel.

rdkbRgL2ogreSecRemoteAddressType

The address type (IPv4 or IPv6) for the secondary remote endpoint of the GRE tunnel.

rdkbRgL2ogreSecRemoteAddress

The IP address for the secondary remote endpoint of the GRE tunnel.

4. Make other configuration changes in the **rdkbRgWifiHotspotTable**, **rdkbRgL2ogreBase**, and the **rdkbRgL2ogreSourceIfTable** as needed for your network requirements.
5. If you made these changes in the configuration file, restart the router to allow the provisioning to take effect.

Guest SSID

This release of Touchstone firmware supports subscriber-controlled Guest SSIDs for Gateway products. Guest networks support both 2.4 GHz and 5 GHz networks. The primary home SSID and guest SSIDs can have different settings for AP isolation, firewall, parental controls, and similar features.

To enable guest networks:

1. Set the **rdkbRgDot11ExtMbssUserControl** object (.32 for 2.4 GHz and .112 for 5 GHz) to designate which SSIDs are available as guest networks. You can either use a bit mask to designate arbitrary SSIDs, or set a value 2 through 8 to enable the private network plus up to 7 guest networks.
2. For each SSID under subscriber control, set the **rdkbRgDot11ExtOperMode.x** object to **local** (3). The index x is the **ifIndex** of each BSS.
3. Set the **rdkbRgDot11ApplySettings** object to **true**(1) to apply the settings.

Band Steering

AR01.1 supports Band Steering, a method procedure for automatically associating Wi-Fi clients between a pair of 2.4 GHz and 5 GHz SSIDs.

Band Steering Operation

If a client sends a broadcast probe request:

1. If the probe request is on the 2.4 GHz band, continue normal operation.
2. If the probe request is on the 5 GHz band, and the client is already associated:
 - a. If the client RSSI is above a configurable threshold, add the client to a 2.4 GHz blacklist table (if it is not already in the list), forcing it to connect to the 5 GHz SSID.
 - b. If the RSSI is below the threshold, remove the client from the 2.4 GHz blacklist table (if it is in the list).
 - c. Add the client to the 5 GHz capable list (if it is not already in the list), then continue normal operation.
3. If the probe request is on the 5 GHz band, and the client is not associated:
 - a. Add the client to a list of 5 GHz-capable clients, if necessary.
 - b. If the client RSSI is above a configurable threshold, add the client to a 2.4 GHz blacklist table, forcing it to connect to the 5 GHz SSID.
 - c. If the RSSI is below the threshold, remove the client from the blacklist and allow it to connect to the 2.4 GHz SSID.

If a client sends a direct probe request:

1. If the probe request is on the 2.4 GHz band:

- a. Check the list of 5 GHz-capable clients. If the client is not in that list, continue normal operation.
 - b. If the client is listed as 5 GHz-capable, check the 2.4 GHz blacklist. If it is, ignore the request; otherwise, continue normal operation.
2. If the probe request is on the 5 GHz band, and the device is already associated:
 - a. If the client RSSI is above a configurable threshold, add the client to a 2.4 GHz blacklist table, forcing it to connect to the 5 GHz SSID.
 - b. If the RSSI is below the threshold, remove the client from the blacklist and allow it to connect to the 2.4 GHz SSID.
3. If the probe request is on the 5 GHz band, and the device is not associated:
 - a. If the client RSSI is above a configurable threshold, add the client to a 2.4 GHz blacklist table, forcing it to connect to the 5 GHz SSID.
 - b. If the RSSI is below the threshold, ignore the probe request.

Band Steering Requirements

For Band Steering to work, both a 2.4 GHz SSID and a 5 GHz SSID must be enabled. The SSIDs must have identical:

- names
- encryption types
- password/pre-shared keys

Band Steering is configured through TR-069.

Configuring Band Steering

To configure Band Steering:

1. Make sure you have SSIDs configured properly, as described in "Band Steering Requirements" above.
2. Enable Band Steering on both SSIDs, using TR-069. For example, if you are enabling Band Steering on the subscriber's private network:
`Device.WiFi.X_ARRIS_COM_BandSteering.SSID.10001.Enable = True`
`Device.WiFi.X_ARRIS_COM_BandSteering.SSID.10101.Enable = True`
3. Make other changes as necessary to the Band Steering parameters, or accept the defaults. See [Band Steering Parameters \(AR01.1\)](#) (page 368) for a list of configurable parameters.
4. Apply the settings. This resets the radios; there may be a noticeable delay.

ARRIS-ROUTER-MIB Objects Supported in AR01.1

The following router objects are supported in this release. See below for other objects with RDK-MIB equivalents.

arrisRouterDSLiteWanEnable

Set to **true**(1) to enable DS-Lite on this Gateway.

arrisRouterDSLiteTcpMssClamping

Set to **true**(1) to enable MSS clamping for the IPv6 tunnel.

arrisRouterDSLiteTcpMssValue

The TCP MSS value for the IPv6 tunnel.

arrisRouterRDKIPv4LanClientTable

Information about each IPv4 client associated to the eRouter.

arrisRouterRDKIPv6LanClientTable

Information about each IPv6 client associated to the eRouter.

The following objects in the **arrisRouterBSSTable** are supported. The index for this table is the ifIndex of the BSS.

arrisRouterBssID

The MAC address of the Gateway.

arrisRouterBssSSID

The SSID for this BSS.

arrisRouterBssActive

Set to **true**(1) to make this SSID active.

arrisRouterBssSSIDBroadcast

Set to **true**(1) to broadcast the SSID.

arrisRouterBssSecurityMode

The security mode of this SSID.

arrisRouterBssNetworkIsolate

Set to **true**(1) to enable network isolation for this SSID. When network isolation is enabled, clients connecting to this SSID have full access to the Internet, but not to devices on the local network.

arrisRouterBssMaxWifiClients

The maximum number of concurrent Wi-Fi clients allowed on this BSS, or 0 for no limits.

arrisRouterBssWmmEnable

Set to **true**(1) to enable WMM (Wi-Fi Multimedia) on this SSID.

arrisRouterBssWmmAPSD

Set to **true**(1) to enable WMM APSD (Wi-Fi Multimedia Automatic Power Save Delivery) on this SSID.

arrisRouterDefaultBssSSID

The factory-default SSID name for this BSS.

The following **arrisRouterWiFi50RadioSettings** objects are supported.

arrisRouterWiFi50BlockDFSChan

Set to **true**(1) to block auto-selection of DFS channels (the subscriber can manually select them).

arrisRouterWiFiMsoEnable

Set to **true**(1) to enable MSO configuration.

arrisRouterWiFiApplySettings

Set to **true**(1) to apply settings to the Gateway configuration.

Supported **arrisRouterFWCfg** objects include:

arrisRouterFWSecurityLevel

The firewall security level; one of: **minimum**(1), **medium**(2), **maximum**(3), **custom**(4), or **none**(5).

arrisRouterFWRemoteMgmtHttp

Set to **true**(1) to enable WAN access to the Gateway web pages.

arrisRouterFWRemoteMgmtHttps

Set to **true**(1) to enable HTTPS access to the Gateway pages from the WAN.

arrisRouterFWVirtSrvTableEnabled

Set to **true**(1) to enable port forwarding.

arrisRouterFWIPv6SecurityLevel

The IPv6 firewall security level; one of: **minimum**(1), **medium**(2), **maximum**(3), **custom**(4), or **none**(5).

arrisRouterFWVirtSrvStatus

The port forwarding status.

The following **arrisRouterSysCfg** objects are supported:

arrisRouterAdminPassword

The administrative password to log into the router web pages.

Mapping ARRIS-ROUTER-MIB Objects to RDK-MIB Objects

The following table lists ARRIS-ROUTER-MIB objects with equivalents in the RDK-MIB. Only supported objects are included.

ARRIS-ROUTER-MIB Objects	RDK-B MIB Objects
arrisRouterWPAPreSharedKey	rdkbRgDot11WpaPreSharedKey
arrisRouterBssSSID	rdkbRgDot11BssSsid
arrisRouterWiFiMode	rdkbRgdot11nExtMode

ARRIS-ROUTER-MIB Objects	RDK-B MIB Objects
arrisRouterWiFiChannel	clabWIFIRadioChannel
arrisRouterWiFi50Mode	rdkbRgdot11nExtMode
arrisRouterWiFiEnableRadio	clabWIFIRadioEnable
arrisRouterWiFi50Enable	clabWIFIRadioEnable.2
arrisRouterWiFiChannelBW	clabWIFIRadioOperatingChannelBandwidth
arrisRouterWiFi50ChannelBW	clabWIFIRadioOperatingChannelBandwidth.2
arrisRouterWiFiPhysicalChannel	clabWIFIRadioChannelsInUse
arrisRouterWiFi50PhysicalChannel	clabWIFIRadioChannelsInUse.2
arrisRouterWiFi50Channel	rdkbRgDot11ExtCurrentChannel
arrisRouterBssSecurityMode	rdkbRgDot11BssSecurityMode
arrisRouterFWSecurityLevel	rdkbRgFirewallProtection
arrisRouterRDKLanClientTable	rdkbRgIpMgmtLanConnectedClientsTable
arrisRouterWiFiClientInfoRSSI	rdkbRgIpMgmtLanConnectedClientsRSSI
arrisRouterRDKIPv4LanClientMAC	rdkbRgIpMgmtLanConnectedClientsPhysAddr
	rdkbRgIpMgmtLanConnectedClientsAddressSource
arrisRouterRDKIPv4LanClientIPAddr	rdkbRgIpMgmtLanConnectedClientsIpv4Addr
arrisRouterRDKIPv4LanClientHostName	rdkbRgIpMgmtLanConnectedClientsHostName
arrisRouterRDKIPv4LanClientAdapterType	rdkbRgIpMgmtLanConnectedClientsInterface
arrisRouterLanClientOnline	rdkbRgIpMgmtLanConnectedClientsActive
	rdkbRgIpMgmtLanConnectedClientsComments
arrisRouterLanPassThru	rdkbRgIpMgmtLanMode
arrisRouterBssWmmEnable	rdkbRgDot11ExtWmm

The following ARRIS-ROUTER-MIB objects are not supported and have no RDK-MIB equivalent:

- mocaNodeMocaVersion
- mocaNodeMacAddress
- mocaNodeTxGcdRate
- mocaNodeRxPower
- arrisRouterWiFi50HTMode
- arrisRouterLanClientCount
- arrisRouterLanClientType

- arrisRouterLanClientRowStatus
- arrisRouterLanEtherPortIFIndex
- arrisRouterLanEtherPortEnabled
- arrisRouterLanEtherPortDuplex
- arrisRouterLanEtherPortSpeed
- arrisRouterLanEtherPortAuto
- arrisRouterLanEtherPortHasLink

Operations

Operations encompass automatic monitoring of environment, detecting and determining faults and alerting administrators of problems.

Battery Management

Touchstone firmware provides a sophisticated management and monitoring scheme to maximize battery hold times and extend the useful life of backup batteries.

Initial Battery Charging

When the E-UE is powered up, whether for the first time or after replacing a battery, it begins a battery charging and testing sequence. See the *Touchstone Battery Reference Manual* for details about the charging and testing sequence.

Battery Telemetry

Touchstone firmware provides battery telemetry through the following management interfaces:

- LED display
- SNMP

Power Failure Operation

When an E-UE with battery backup capabilities loses AC power, it immediately takes the following actions:

1. Shuts off some LEDs to conserve battery power.
2. Shuts down the data services running over all LAN ports after 30 seconds (default) of power loss. You can change the amount of time before the E-UE disables data services, by setting the [arrisMtaDevPwrSupplyDataShutdownTime](#) object.
3. After the shutdown timer expires, E-UEs disable bonded mode, switching to 1x1 unbonded (see ["Advanced Power Management"](#) (page 158) for details).

LED Changes

LED behavior changes under specific conditions, as described below.

LED Operation Changes During Battery Charging

The Battery LED only flashes when AC power is not present and the battery is low, or else when the battery needs to be replaced. The Battery LED does not flash during normal charging.

Installer Visual Indication of Foreign Loop Voltage

LEDs flash any time the line card transitions to the Line Card Over-current Protection State. This indicates foreign voltage is present on the loop. This condition occurs most often at installation time.

Battery Status Monitoring

Touchstone firmware provides battery status monitoring through SNMP. Battery MIB objects provide an estimate of remaining battery charge as a percentage of full charge, and in minutes remaining to depletion.

E-UEs have battery charger hardware that reports an accurate estimate of battery capacity within 5 minutes of initialization.

The "Power Supply Telemetry" alarms and logs provide a report of any status changes to the power system, including the battery and the charger.

Highest Charger Temperature Recording

Touchstone firmware provides a feature to record and report the historic high charger temperature:

arrisMtaDevPwrSupplyHighestTemperature

Provides the highest temperature, in degrees C, recorded by the battery charger.

arrisMtaDevPwrSupplyHighestTemperatureTime

Provides the date and time that the E-UE recorded the high temperature.

arrisMtaDevPwrSupplyHighestTemperatureClear

Resets the highest temperature and time data.

Advanced Power Management

Touchstone E-UEs with battery backup support enhanced low-power features to maximize battery hold-up time.

When the E-UE loses AC power, it starts the data shutdown timer with a default value of 30 seconds. If the timer expires before AC power is restored, the device:

- powers down Ethernet and wireless (if equipped) interfaces.
- switches to 1x1 unbonded mode after all lines go on-hook, informing the CMTS using a CM-STATUS message.

Once AC power is restored, the E-UE:

- activates data interfaces (as provisioned)
- attempts to restore its original bonding mode
- clears codeword counters as a side effect

CMTS Considerations

For best results, the CMTS should fully support CM-STATUS partial service messages. If the CMTS does not support CM-STATUS, add the [arrisCmDoc30SetupPowerSaveMode](#) object to the CM configuration file with a value of `reinitmac(1)`. This causes the CM to re-register in unbonded (1x1) mode upon loss of AC power, and re-register to bonded mode when AC power is restored.



Note: Re-registration may take up to 90 seconds.

When the CMTS is upgraded, change the value to the default `partial service(0)`, or remove the object from the CM configuration file.

About IPv6 Support

IPv6 allows cable operators to expand their IP address ranges for cable modems, allowing reclamation of traditional IPv4 addresses for use with eDVAs. AR01.1 supports the following IPv6-related protocols:

- DHCPv6
- DNSv6
- TFTPv6
- TODv6
- SNMPv6
- TELNETv6
- HTTPv6
- SSHv6



Note: Touchstone firmware does not support Early Authentication and Encryption (EAE) functionality.

Supported Hardware

Touchstone firmware supports single- and dual-stack CPE traffic on all supported Touchstone devices as follows:

- supports single stack IPv4 traffic to and from CPEs.
- supports single stack IPv6 traffic to and from CPEs.
- supports IPv4 and IPv6 traffic simultaneously to and from CPEs.

Touchstone firmware supports IPv6 operation on all supported Data Gateway and Telephony Gateway products.

IPv6 Multicast Support

AR01.1 supports multicast forwarding as described below.

Before Registration

The CM:

- filters multicast traffic not addressed to the IPv6 Link Local Scope All Nodes Address or the Solicited Node Addresses.
- learns the pre-registration DSID from the MDD message.
- forwards pre-registration multicast traffic to its IP stack based on the pre-registration DSID as defined in the "DSID based Filtering and Forwarding by a Cable Modem" section of CM-SP-MULPIv3.0-I12-100115.

During Registration

The CM reports the following IPv6 multicast capabilities in DHCP Option 125:

Sub-Option	Description	Value
32	Multicast DSID support	24
33	Multicast DSID forwarding (GMAC promiscuous)	2
34	Frame Control Type forwarding	1

After Registration

The CM:

- stops forwarding multicast traffic labeled with the Pre-Registration DSID, after receiving the REG-RSP message.

- forwards multicast traffic based on the DSIDs and Group MAC Addresses received in the REG-RSP message, as specified by the "DSID based Filtering and Forwarding by a Cable Modem" section of CM-SP-MULPIv3.0-I12-100115.
- filters multicast traffic based on DSIDs and Group MAC Addresses received in the REG-RSP message. The CMTS always includes the IPv6 Link Local Scope All Nodes Address, and the CM's Solicited Node Addresses, in the REG-RSP message.
- supports Multicast DSID Encodings (TLV-50.4) described in the "Multicast Encodings" section of CM-SP-MULPIv3.0-I12-100115, as well as Security Association Encodings (TLV-51) described in the "Security Association Encoding" section of CM-SP-MULPIv3.0-I12-100115, received in the REG-RSP message.
- uses the Dynamic Bonding Change mechanism defined in CM-SP-MULPIv3.0-I12-100115, to maintain and learn new DSID values and Security Associations for IPv6 multicast.

The CM also forwards Neighbor Discovery packets sent to:

- The CPEs' Solicited Node multicast IPv6 addresses
- The All Nodes multicast address onto CPE ports

Neighbor Discovery forwarding can be controlled by configuring filters to allow or prohibit forwarding.

IPv6 Management

Managing Touchstone E-UEs using IPv6 addressing in the CM, and IPv4 addressing in the eDVA, require SNMP management software capable of handling both IPv4 & IPv6, or separate tools for CM and eDVAs.

Filtering IPv6 Traffic

AR01.1 supports the following filtering-related features:

- MLD Snooping
- Upstream Drop Classifiers

Each of these features is described below.

MLD Snooping

MLD Snooping allows CPE devices using either MLDv1 or MLDv2 to acquire IPv6 addresses, and to register and receive multicast IPv6 downstream traffic. Unless filters specifically prohibit it, MLD Snooping allows all multicast IPv4 and IPv6 traffic to pass.



Note 1: BPI treats IPv6 multicast traffic like normal traffic (IPv4 multicast traffic is treated as multicast).

Note 2: In AR01.1, MLDv2 Source Filtering is ignored.

When the CM receives a DOCSIS Reinit-MAC command, it clears the MLD database and downstream IPv6 multicast filtering records.

Upstream Drop Classifiers

An Upstream Drop Classifier is a Classifier created by the CM to filter upstream traffic. The CM performs IP protocol filtering using either Upstream Drop Classifiers or IP filters. Traffic may also be filtered from the WAN to the LAN, set by the user.

If a packet matches the specified packet-matching criteria of an Upstream Drop Classifier, it is dropped. Unlike QoS Classifiers, Upstream Drop Classifiers do not refer to a Service Flow.

TCP/UDP Packet Classification Encodings are defined for IPv4 or IPv6 and may be present in a Service Flow Classifier of either type. If those classifiers are present in combination with IPv6 classifier encodings, they apply to the IPv6 classifiers.

The CM reports the number of Upstream Drop Classifiers supported, using the Upstream Drop Classification Support capability (TLV 5.38) in the Registration Request message.

Coexistence

AR01.1 supports SNMPv3 co-existence for both the ARRIS version of coexistence and coexistence using the coexistence MIBs.

The following is an example provisioning file fragment for setting up co-existence.

```
SnmpMib = snmpCommunityStatus.rocablelabs.createAndGo
SnmpMib = snmpCommunityName.rocablelabs "ro_cm"
SnmpMib = snmpCommunitySecurityName.rocablelabs "rotesting1"
SnmpMib = snmpCommunityStorageType.rocablelabs volatile
SnmpMib = snmpCommunityStatus.rwcablelabs.createAndGo
SnmpMib = snmpCommunityName.rwcablelabs "rw_cm"
SnmpMib = snmpCommunitySecurityName.rwcablelabs "rwtesting1"
SnmpMib = snmpCommunityStorageType.rwcablelabs volatile
SnmpMib = vacmSecurityToGroupStatus.1.rotesting1.createAndGo
SnmpMib = vacmGroupName.1.rotesting1 "rotesting2"
SnmpMib = vacmSecurityToGroupStorageType.1.rotesting1 volatile
SnmpMib = vacmSecurityToGroupStatus.1.rwtesting1.createAndGo
SnmpMib = vacmGroupName.1.rwtesting1 "rwtesting2"
SnmpMib = vacmSecurityToGroupStorageType.1.rwtesting1 volatile
SnmpMib = vacmSecurityToGroupStatus.2.rotesting1.createAndGo
SnmpMib = vacmGroupName.2.rotesting1 "rotesting2"
SnmpMib = vacmSecurityToGroupStorageType.2.rotesting1 volatile
SnmpMib = vacmSecurityToGroupStatus.2.rwtesting1.createAndGo
SnmpMib = vacmGroupName.2.rwtesting1 "rwtesting2"
SnmpMib = vacmSecurityToGroupStorageType.2.rwtesting1 volatile
SnmpMib = vacmAccessStatus.rotesting2 1 1.createAndGo
SnmpMib = vacmAccessContextMatch.rotesting2 1 1 exact
SnmpMib = vacmAccessReadViewName.rotesting2 1 1 "docsisManagerView"
SnmpMib = vacmAccessStorageType.rotesting2 1 1 volatile
SnmpMib = vacmAccessStatus.rwtesting2 1 1.createAndGo
SnmpMib = vacmAccessContextMatch.rwtesting2 1 1 exact
SnmpMib = vacmAccessReadViewName.rwtesting2 1 1 "docsisManagerView"
SnmpMib = vacmAccessWriteViewName.rwtesting2 1 1 "docsisManagerView"
SnmpMib = vacmAccessStorageType.rwtesting2 1 1 volatile
SnmpMib = vacmAccessStatus.rotesting2 2 1.createAndGo
SnmpMib = vacmAccessContextMatch.rotesting2 2 1 exact
SnmpMib = vacmAccessReadViewName.rotesting2 2 1 "docsisManagerView"
SnmpMib = vacmAccessStorageType.rotesting2 2 1 volatile
SnmpMib = vacmAccessStatus.rwtesting2 2 1.createAndGo
SnmpMib = vacmAccessContextMatch.rwtesting2 2 1 exact
```

```

SnmpMib = vacmAccessReadViewName.rwtesting2 2 1 "docsisManagerView"
SnmpMib = vacmAccessWriteViewName.rwtesting2 2 1 "docsisManagerView"
SnmpMib = vacmAccessStorageType.rwtesting2 2 1 volatile
SNMPV3Kickstart =
    SNMPV3SecurityName = "docsisManager"
    SNMPV3PublicNumber = hexstr: C1. FC. 52. 36. 97. 06. C0. 22. 99. 61. D5. CA. C5.
4F. C6. 68. 51. 71. A8. DC. 69. AB. EB. D6. 21. AC. AC. 1D. FC. A6. 0A. 3A. 8E. 77. B5. 15. AB.
AC. 60. 7C. 5F. EB. AF. 5F. 86. B8. 3F. 2B. A1. DB. 3D. ED. 51. E2. EB. 5D. E0. 6A. EB. 2D. AE.
E3. A4. DA. AC. DA. 30. 42. DC. A2. 3C. 5B. FE. 65. 83. B8. B8. 9E. 48. 02. FB. 70. A5. E9. 97.
0C. 95. 9F. 96. 44. B4. BA. B4. 2C. 71. 97. D1. 1A. 96. 99. C9. 4F. 9C. 53. 3F. 00. 24. 3E. 1A.
12. AB. 23. CF. DB. 05. 6C. 97. 62. 4B. B2. A3. FC. 7D. 91. F4. 90. C7. 7C

```

DHCPv6 MIB Objects

The following MIB objects indicate the DHCP mode (v4 or v6) and assigned IP address. The address type for each object is one of the following:

unknown(1):

The address type is unknown

ipv4(1):

DHCPv4

ipv6(2):

DHCPv6

ipv4z(3):

IPv4 non-global address with a zone index

ipv6z(4):

IPv6 non-global address with a zone index

dns(16):

DNS domain name

The objects are:

docsDevServerDhcpAddressType

The IP address type of the assigned address, or **unknown(0)** if the IP address was statically assigned.

docsDevServerDhcpAddress

The IP address assigned to the CM, or an empty string if the IP address was statically assigned.

The following MIB objects indicate the IP address and type of the assigned time and TFTP servers:

docsDevServerTimeAddressType

The IP address type for the time server.

docsDevServerTimeAddress

The IP address of the time server.

docsDevServerConfigTftpAddressType

The IP address type of the TFTP server.

docsDevServerConfigTftpAddress

The IP address of the TFTP server.

The Syslog server address must be configured manually, either through an SNMP manager or the configuration file. In either case, the following two objects must be set in the order shown:

docsDevEvSyslogAddressType

The IP address type; either **1** (IPv4) or **2** (IPv6).

docsDevEvSyslogAddress

The IP address of the Syslog server, or a blank string to disable Syslog transmission.

The following read-only MIB objects are part of **arrisCmDoc30DhcpCmParameters**, and provide DHCP information.

arrisCmDoc30DhcpCmIpAddrType

The type of the currently leased IP address.

arrisCmDoc30DhcpCmIpAddr

The currently leased IP address.

arrisCmDoc30DhcpCmSubNetMask

The current IP subnet mask in use.

arrisCmDoc30DhcpCmGatewayIpAddr

The current IP gateway address in use.

arrisCmDoc30DhcpCmConfigFile

The CM configuration file name.

SNMP Access

The CM expects SNMP access to use the same IP mode (IPv4 or IPv6) that it is provisioned with.

Event Reporting

AR01.1 supports PacketCable 1.0 event reporting functionality and a number of proprietary ARRIS events. The following enterprise numbers may appear in events generated by Touchstone E-UEs:

Number	Source
4115	ARRIS
4491	PacketCable

Collecting Events

Events may be collected by:

- The **docsDevEventTable** (part of the DOCS-CABLE-DEVICE-MIB) in the E-UE keeps CM events until the E-UE is rebooted or powered down.
- The **pktcDevEventTable** (part of the PKTC-EVENT-MIB) in the E-UE keeps eDVA events until the E-UE is rebooted or powered down. The default configuration stores events only in the local event tables.
- A specified Syslog server (specify a Syslog server using the **pktcDevEvSyslogAddress** object).
- SNMP servers (the E-UE sends the events in a **pktcDevEventNotify** message). See *Event Formats* (page 165) below for details.

Event Formats

AR01.1 firmware provides log messages for both the cable modem and eDVA sections of the Touchstone E-UE.

Cable Modem Log Format

Cable modem logs require the DOCS-CABLE-DEVICE-MIB. Cable modem log messages consist of the following information:

- **EventIndex**—Provides relative ordering of the objects in the event log. The value of this object always increases except when:
 - the log is reset using the **docsDevEvControl** object
 - the device reboots and does not implement non-volatile storage for this log
 - the value reaches 2^{31} (the index is a 32-bit counter that rolls over to zero at this limit)The next value for all the above cases is 1.
- **EventFirstTime**—The time that this entry was created.
- **EventLastTime**—If multiple events are reported via the same entry, the time that the last event for this entry occurred, otherwise this should have the same value as **EventFirstTime**.
- **EventCounts**—The number of consecutive event instances reported by this entry. This starts at 1 with the creation of this row and increments by 1 for each subsequent duplicate event.
- **EventLevel**—The priority level of this event as defined by the vendor. These are ordered from most serious (emergency) to least serious (debug).
- **EventId**—For this product, uniquely identifies the type of event that is reported by this entry.
- **EventText**—Provides a human-readable description of the event, including all relevant context.

eDVA Log Format

eDVA logs function within the context of the PKTC-EVENT-MIB. eDVA log messages consist of the following information:

Event Index

Provides relative ordering of the objects in the event log. This also serves as a indicator of event sequence. The object value always increases except when:

- the log is reset using the **pktcDevEvControl** object
- the device reboots and does not implement non-volatile storage for this log
- the value reaches 2^{31} (the index is a 32-bit counter that rolls over to zero at this limit)

The next entry for all the above cases is 1.

Event Time

Provides a human-readable description of the time at which the event occurred.

Event Level

The priority level of this event as defined by the vendor.

Event Enterprise number

The IANA enterprise number: **4115** for ARRIS events, **4491** for PacketCable events.

Event ID

ID for a specific event to which the priority and display string are matched. Event IDs are vendor specific.

Event Text

The text message associated with the event. Corresponds to the **pktcDevEvProgrammableId** or **pktcDevEvFixedId** MIB objects.

Mac Address

Provides the MAC address of the device generating the event.

FQDN/Endpoint ID

The FQDN or IP address of the device, with an endpoint identifier. If the event is not specific to an endpoint, the identifier is simply the FQDN or IP address. If the event is specific to an endpoint, the format is *AALN/X: FQDN/IP address*. If the event is battery related, the format is *FQDN/IP address: BATT_PORT: 1*.



Note: If the **arrisMtaDevEventHideMacFQDNandIPAddress** object is set to **enable(1)**, the endpoint FQDN and IP address are masked with asterisk (*) characters as described below.

References

The following CableLabs® specifications define the PacketCable 1.0 event reporting mechanism:

PKT-SP-MEM1.5-I03-070412

PacketCable Management Event Mechanism

PKT-SP-EVEMIB1.5-I02-050812

PacketCable Management Event MIB Specification

PKT-TR-MEMEVENT-ID-V01-0000929

PacketCable Management Event Identifiers

Event Summary

Touchstone firmware generates events (alarms and log messages) from both the CM and eDVA components. This manual documents only ARRIS-specific events; for DOCSIS CM events, see the *DOCSIS 3.0 Operations Support System Interface Specification*, CM-SP-OSSlv3.0-I15-100115.

Event Handling

Events may be collected at:

- The **docsDevEventTable** (part of the DOCS-CABLE-DEVICE-MIB) in the E-UE keeps CM logs until the E-UE is rebooted or powered down.
- The **pktcDevEventTable** (part of the PKTC-EVENT-MIB) in the E-UE keeps eDVA logs until the E-UE is rebooted or powered down. The default configuration stores logs only in the local event tables.
- A specified Syslog server (specify a Syslog server using the **pktcDevEvSyslogAddress** object).
- SNMP servers (the E-UE sends the log in a **pktcDevEventNotify** message).

eDVA States

The following is a list of valid eDVA states and their meanings. The **ppSurvMtaMaintState** object contains the current state.

In Service, Normal

The eDVA is operating normally.

In Service, Trouble

The eDVA is providing service, but a problem has been detected (typically loss of AC power).

Out of Service

The eDVA is out of service due to a detected problem. Both voice and data services may be affected.

eDVA Line States

The following is a list of valid eDVA line states and their meanings. The value of the **ppSurvPortMaintState** object provides the current line state.

In Service, Normal

The line is operating normally.

In Service, Trouble, Family Equipment Failure

A problem with the eDVA subsystem is preventing the line from functioning properly. All lines in this state may indicate that the eDVA failed to download its firmware image.

In Service, Trouble, Test Failed

The line failed diagnostics. The *Voice Line Diag Failed* (page 171) log provides more details.

In Service, Trouble, Diagnostics

The line is running diagnostics; if no problems are discovered, the line will be returned to service when diagnostics are finished.

In Service, Trouble, Line Card Protection

The line card is in an overcurrent protection state. This state usually indicates either a short between tip and ring, or a foreign voltage being applied to tip and ring.

Out of Service, Normal, Unprovisioned

The line is not provisioned but has no known problems.

Out of Service, Normal

The line card is out of service and provisioned, but has no known problems.

Out of Service, Trouble

The line is out of service due to a detected problem.

Out of Service, Trouble, Diagnostics

The line is out of service, and is running diagnostics.

E-UE Battery States

The following is a list of valid battery states for E-UEs with battery backup. The value of the **arrisMtaDevBatteryOperState** object provides the current battery state.

normal (12)

The E-UE is operating on AC power. The battery is charged and in good condition.

acFail (11)

The E-UE is operating on battery power.

batteryLow(7)

The E-UE is operating on AC power, but during a power outage has drawn down the battery to the capacity indicated by the **arrisMtaDevPwrSupplyLowBatteryThresh** object.

batteryLow-replaceBattery(5)

The E-UE is operating on AC power. However, in addition to the Battery Low condition described above, the battery has deteriorated and should be replaced.

shutdownWarning(2)

The E-UE has nearly exhausted its battery power, and will lose power if AC power is not restored within a few minutes.

batteryMissing(8)

The battery has been removed or has failed in such a way to appear to be removed.

replaceBattery(10)

The E-UE is operating on AC power. However, the battery has deteriorated and should be replaced.

unavailable(0)

The E-UE does not support battery telemetry.

invalid(1)

Indicates a possible problem with the E-UE or the battery system.

batteryReversedShorted(3)

The battery has either been installed backwards or the terminals have been shorted.

batteryLow-acFail(6)

The E-UE is operating from battery power, and has entered the “Battery Low” condition described above.

acFail-replacebattery(9)

The E-UE is operating from battery power. In addition, the battery has deteriorated as described in “Battery Replace” above and should be replaced.

batteryLow-replaceBattery-acFail(4)

The E-UE is operating from battery power, and has entered the “Battery Low” condition described above. In addition, the battery has deteriorated as described in “Battery Replace” above and should be replaced.

testInProgress(13)

The E-UE is testing the battery and charger system.

chargerFailure(14)

The E-UE has twice failed (initial attempt and one retry) to download charger firmware. This indicates a hardware problem with the E-UE. The battery charger is disabled and backup battery power is not available.

eDVA Event Summary

The following is a list of ARRIS and PacketCable eDVA events that the AR01.1 firmware can generate. See the section describing the event for further details.

Ent.	ID	Severity	Log Text
4115	1	information(5)	Voice Line Diag Failed
4115	2	information(5)	Voice Line Diag Passed
4115	3	information(5)	Voice Line State Change
4115	4	information(5)	Voice Line Provisioning Complete
4115	6	information(5)	Voice Line Protection State Change
4115	10	information(5)	State Changed
4115	14	information(5)	MTA TFTP: Failed
4115	16	information(5)	MTA TFTP: Successful
4115	26	information(5)	MTA PROV: Successful!
4115	31	information(5)	Loop Voltage Management: Policy Missing
4115	32	information(5)	Loop Voltage Management: Bad Key
4115	33	information(5)	Loop Voltage Management: Policy Out of Range
4115	34	information(5)	Loop Voltage Management: Policy Change
4115	35	information(5)	Loop Voltage Management: Policy 3 Timer Out of Range; Default Value Used
4115	37	information(5)	Call Stats
4115	39	information(5)	Last NCS Message Received
4115	65519	information(5)	Power Supply Telemetry Alarm - Battery Missing
4115	65520	information(5)	Power Supply Telemetry Alarm - Battery Low
4115	65521	information(5)	Power Supply Telemetry Alarm - Replace Battery
4115	65523	information(5)	SIP General Failure
4115	65524	information(5)	SIP Network Failure
4115	65526	information(5)	SIP Authentication Failure
4115	65527	information(5)	SIP Registration Timeout
4115	65528	information(5)	SIP Proxy Loss of Communications
4115	65533	major(2)	Voice Line Failure
4115	46	information(5)	MTA DHCP RENEW: Lease Renewal delay; Voice line offhook
4115	47	information(5)	MTA DHCP REBIND: Lease Renewal delay; Voice line offhook
4115	65529	major(2)	Power Supply Telemetry Alarm
4115	2417164301	information(5)	SSH LOGIN ACCEPTED
4115	2417164302	information(5)	SSH LOGIN REJECTED
4115	2417164303	information(5)	SSH LOGIN REJECTED - MAX ATTEMPTS
4115	2417164296	information(5)	Touchstone SW Upgrade Failed Before Download Attempt
4115	2417164297	information(5)	Touchstone SW Upgrade Failed
4115	2417164298	information(5)	Touchstone SW Upgrade Successful

Ent.	ID	Severity	Log Text
4115	1	information(5)	Voice Line Diag Failed
4115	2417164299	minor(3)	Touchstone SW Upgrade Aborted due to Battery AC-FAIL Condition
4115	2417164304	minor(3)	Touchstone SW Upgrade Aborted due to Call in Progress
4115	2417164305	information(5)	Touchstone SW Upgrade Reboot Delayed due to Call in Progress
4115	2417164308	information(5)	Gateway has reset
4115	2417164309	information(5)	Unit has been restored to factory defaults
4491	65528	minor(3)	Battery Not Low
4491	65529	minor(3)	Battery Low
4491	65530	minor(3)	Battery Present
4491	65531	minor(3)	Battery Missing
4491	65532	minor(3)	Battery Good
4491	65533	minor(3)	Replace Battery
4491	65534	minor(3)	AC Restored
4491	65535	minor(3)	AC Fail

ARRIS Events

The following are ARRIS eDVA-related events. ARRIS events use the enterprise number **4115**.

Voice Line Diag Failed

The eDVA has failed manual diagnostics for the specified line.

Format:

Voice Line Diag Failed, Line Number = *line*, Failure Reason = *reason*

Fields:

The fields are as follows:

- *line*—The line number that failed diagnostics. The first line is line 1.
- *reason*—one of the following:
 - Line is Unprovisioned
 - Invalid State to Init Diags
 - Power/Clock Failure
 - SLAC Revision Failure
 - MPI Failure
 - PCM Failure
 - Standby Hook Failure
 - Active Hook Failure

- VF Failure
- Ringing Failure

Action:

Use the reason code to determine the course of action as follows:

- Line is Unprovisioned—Provision the line and re-try the diagnostics.
- Invalid State to Init Diags—Set the line state to **oos** and re-try the diagnostics.
- others—Reset the eDVA and re-try diagnostics. If the problem persists, replace the E-UE.

Voice Line Diag Passed

Indicates that the eDVA has successfully completed manual diagnostics on the specified line.

Format:

Voice Line Diag Passed, Line Number = *line*

Fields:

line indicates the line that passed diagnostics. The first line is line 1.

Action:

None.

Voice Line State Change

The eDVA received an operator-requested state change for the specified line.

Format:

Voice Line State Change, Line Number = *line*, Prev State = *old_state*, New State = *new_state*

Fields:

The fields are as follows:

- *line*—the line number that changed state. The first line is line 1.
- *old_state*, *new_state*—the previous and current line states; see [eDVA Line States](#) (page 168) for details.

Action:

If the new state indicates a trouble condition, correct the problem.

Voice Line Protection State Change

The specified line has detected or cleared a protection fault.

Format:

Voice Line Protection State Change, Line Number = *line*, New State = *new_state*

Fields:

The fields are as follows:

- *line*—the line number that changed state. The first line is line 1.
- *new_state*—the current protection state; one of:
 - Fault DETECTED
 - Fault CLEARED

Action:

Monitor the E-UE for service issues and possible replacement.

Power Supply Telemetry Log

The E-UE has detected a change in its battery telemetry state.

Format:

Power Supply Telemetry - *state*

Fields:

The *state* field reflects the telemetry state; see [E-UE Battery States](#) (page 168) for details. If a Replace Battery condition persists for 24 hours, the E-UE generates another Power Supply Telemetry log.

Action:

Replace the E-UE battery if indicated by the telemetry state.

MTA TFTP: Successful

The E-UE successfully downloaded its eDVA provisioning file.

Format:

MTA TFTP: Successful

Action:

None.

MTA PROV: Successful!

The E-UE successfully completed its eDVA provisioning.

Format:

MTA PROV: Successful!

Action:

None.

SSH LOGIN ACCEPTED

A user has successfully logged into the Telephony Modem using SSH.

Format:

Successful SSH login by *name* from *i paddr* on *date*.

Fields:

The *i paddr* is the IP address (e.g. 192.168.42.42) of the client logging into the Telephony Modem. The *name* field is the user name entered. The *date* field is the date and time of login.

Action:

None.

SSH LOGIN REJECTED

A user has unsuccessfully attempted to log into the Telephony Modem using SSH.

Format:

SSH LOGIN REJECTED FROM IP [*i paddr*] . USERNAME - (*name*).

Fields:

The *i paddr* is the IP address (e.g. 192.168.42.42) of the client logging into the Telephony Modem. The *name* field is the user name entered.

Format:

Failed SSH login attempt from *i paddr* on *date*.

Fields:

The *i paddr* is the IP address (e.g. 192.168.42.42) of the client logging into the Telephony Modem. The *date* field is the date and time of the attempt

Action:

None. However, a large number of unsuccessful login attempts may indicate a potential intrusion attempt.

SSH LOGIN REJECTED - MAX ATTEMPTS REACHED

A user unsuccessfully attempted to log in too many times. The session was disconnected.

Format:

SSH LOGIN REJECTED - MAX ATTEMPTS REACHED

Action:

Continue to monitor the logs for “SSH LOGIN REJECTED” messages. Blocking the IP addresses of any attempted intruders may be necessary to protect your network.

Touchstone Firmware Upgrade Failed Before Download Attempt

Touchstone Firmware Upgrade failed before attempting a download.

Format:

Touchstone Firmware Upgrade Failed Before Download Attempt: *reason*

Fields:

The *reason* description is one of the following:

Provisioned upgrade, bad IP address in [arrisCmDevSwTable](#)

Download initiated via configuration file has failed. A match was found in the ARRIS table but the IP address was invalid. No download was attempted.

Provisioned upgrade, bad filename in [arrisCmDevSwTable](#)

Download initiated via configuration file has failed. A match was found in the ARRIS table but the filename was invalid. No download was attempted.

Provisioned upgrade, no match in [arrisCmDevSwTable](#)

Download initiated via configuration file has failed. No match was found in the ARRIS table. No download was attempted.

Manual upgrade, bad IP address in [arrisCmDevSwTable](#)

Download initiated via remote SNMP browser has failed. A match was found in the ARRIS table but the IP address was invalid. No download was attempted.

Manual upgrade, bad filename in [arrisCmDevSwTable](#)

Download initiated via remote SNMP browser has failed. A match was found in the ARRIS table but the file name was invalid. No download was attempted.

Manual upgrade, no match in [arrisCmDevSwTable](#)

Download initiated via remote SNMP browser has failed. No match was found in the ARRIS table. No download was attempted.

Manual upgrade, device not in operational state

Download initiated via remote SNMP browser has failed. Device not in the operational state. This is also the error returned when [arrisCmDevSwAdminStatus](#) is set to **upgradeFromArri sMgt** in a configuration file.

Action:

Correct the data in the upgrade table or on the TFTP server as necessary, then retry the download.

Touchstone Firmware Upgrade Failed

Touchstone Firmware Upgrade has failed. Standard DOCSIS download logs should also be present to further describe the exact download failure reason.

Format:

Touchstone Firmware Upgrade Failed: *type*

Fields:

The *type* code is one of the following:

Provisioned upgrade

Download initiated via configuration file has failed.

Manual upgrade

Download initiated via remote SNMP browser has failed.

Action:

Make sure the upgrade table and the specified TFTP servers are configured properly, then retry the download.

Touchstone Firmware Upgrade Successful

Touchstone Firmware Upgrade has succeeded. Standard DOCSIS download logs may also be present.

Format:

Touchstone Firmware Upgrade Successful: *type*

Fields:

The *type* code is one of the following:

Provisioned upgrade

Download initiated via configuration file has succeeded.

Manual upgrade

Download initiated via remote SNMP browser has succeeded.

Action:

None.

Touchstone SW Upgrade Aborted due to Battery AC-FAIL condition

Touchstone Firmware Upgrade has failed since the E-UE is running on battery power.

Format:

Touchstone SW Upgrade Aborted due to Battery AC-FAIL condition

Fields:

None

Action:

Wait for AC power to be restored before attempting the firmware upgrade again.

Touchstone SW Upgrade Aborted due to Call in Progress

Touchstone Firmware Upgrade has failed since one or more lines are off-hook and the **arrisMtaDevSwDnldNoSvcImpact** object is set to **StrictEnable(2)**.

Format:

Touchstone SW Upgrade Aborted due to Call in Progress

Fields:

None

Action:

Wait for all lines to go on-hook before attempting the firmware upgrade again.

Touchstone SW Upgrade Reboot Delayed due to Call in Progress

Touchstone Firmware Upgrade has succeeded, but the E-UE has not rebooted since one or more lines are off-hook and the **arrisMtaDevSwDnldNoSvcImpact** object is set to **enable(1)**.

Format:

Touchstone SW Upgrade Reboot Delayed due to Call in Progress

Fields:

None

Action:

None; the E-UE will reboot once all lines to go on-hook.

MTA DHCP RENEW: Lease Renewal delay; Voice line offhook

The eDVA DHCP RENEW sequence has been delayed because the **arrisMtaDevDhcpNoSvcImpact** object is enabled (value is either **dontSend** or **sendIgnore**) and one of the voice lines is offhook.

Format:

MTA DHCP RENEW: Lease Renewal delay; Voice line offhook

Action:

None; the eDVA will not RENEW its IP address until all voice lines are on-hook.

MTA DHCP REBIND: Lease Renewal delay; Voice line offhook

The eDVA DHCP RENEW period has timed out, and the eDVA DHCP REBIND sequence has been delayed, because the [arrisMtaDevDhcpNoSvcImpact](#) object is enabled (value is either **dontSend** or **sendIgnore**) and one of the voice lines is offhook.

Format:

MTA DHCP REBIND: Lease Renewal delay; Voice line offhook

Action:

None; the eDVA will not RENEW its IP address until all voice lines are on-hook.

Power Supply Telemetry Alarm

Severity:

Major

Cause:

The E-UE has lost AC power or has encountered a problem in the battery charging circuitry. The alarm includes one of the following battery status codes:

- AC Fail—the E-UE has detected an AC power failure.
- Replace Battery—the battery has deteriorated to about 75% of its off-the-shelf capacity and should be replaced.
- Battery Missing—the battery was not installed, has been removed, or cannot be detected.

The event log provides more information about the battery status.

Impact:

None at time of alarm. Depending on the condition of the battery and the nature of the power failure, the E-UE may exhaust the battery before AC power is restored.

Action:

Depends on the scope of the power outage.

Gateway has reset

Severity:

Informational

Cause:

The Gateway was reset either through the web pages or by pressing the “Router Reset” button.

Impact:

Data communications (Ethernet or wifi) may be disrupted for a few seconds. Telephony is not affected.

Action:

None.

Unit has been restored to factory defaults

Severity:

Informational

Cause:

The Gateway was restored to factory defaults either through the web pages or by holding the “Router Reset” button for more than 15 seconds.

Impact:

Data communications (Ethernet or wifi) may be disrupted for a few seconds. The subscriber may have to reconfigure the Gateway or restore previous settings. Telephony is not affected.

Action:

None.

Voice Line Provisioning Complete

The eDVA successfully completed provisioning for the specified line.

Event ID: 4**Format:**

Voice Line State Change, Line Number = *line*

Fields:

The fields are as follows:

- *line*—the line number that changed state. The first line is line 1.

Action:

None.

State Changed

The eDVA device changed state, usually as a response to the operator changing the value of the **ppCfgMtaAdminState** object.

Event ID: 10**Format:**

State Changed from *old_state* to *new_state*

Fields:

The fields are as follows:

- *old_state, new_state*—the previous and current device states.

Action:

None.

MTA TFTP: Failed

The device was unable to complete downloading its eDVA file from the TFTP server.

Event ID: 14

Action:

Look for related log messages that may indicate the cause of the problem.

Loop Voltage Management: Policy Missing

The eDVA device has attempted to set up Loop Voltage Management, but the [pktcEnNcsEndPntLVMgmtPolicy](#) object was not configured.

Event ID: 31

Format:

Loop Voltage Management: Policy Missing

Action:

Configure the [pktcEnNcsEndPntLVMgmtPolicy](#) object with the desired policy.

Loop Voltage Management: Bad Key

The eDVA device has attempted to set up Loop Voltage Management, but the [arrisMtaDevLoopVoltageKey](#) object has the wrong value.

Event ID: 32

Format:

Loop Voltage Management: Bad Key

Action:

Configure the [arrisMtaDevLoopVoltageKey](#) object with the correct value. Contact your ARRIS sales support, if necessary, to obtain the correct key value.

Loop Voltage Management: Policy Out of Range

The eDVA device has attempted to set up Loop Voltage Management, but the [pktcEnNcsEndPntLVMgmtPolicy](#) object has the wrong value.

Event ID: 33

Format:

Loop Voltage Management: Policy Out of Range

Action:

Configure the `pktcEnNcsEndPntLVMgmtPolicy` object with a valid policy:

- `vol tageAtAl lTi mes`(1)
- `vol tageUnl essRFQAMabsent`(2)
- `vol tageBasedOnServi ceOrTi mers`(3)
- `vol tageBasedOnServi ce`(4) (default)

Loop Voltage Management: Policy Change

The eDVA device changed policy state.

Event ID: 34

Format:

Loop Voltage Management: Policy Change to *new_policy*

Fields:

The fields are as follows:

- *new_policy*—the new policy value.

Action:

None.

Loop Voltage Management: Policy 3 Timer Out of Range

The eDVA device has attempted to set up Loop Voltage Management using Policy 3, but the `pktcEnNcsEndPntLVMgmtResetTimer` has an improper value and is using the default value of 5 (minutes). This timer applies only when `pktcEnNcsEndPntLVMgmtPolicy` is set to a value of `vol tageBasedOnServi ceOrTi mers`(3).

Event ID: 35

Format:

Loop Voltage Management: Policy 3 Timer Out of Range; Default Value Used

Action:

Configure the `pktcEnNcsEndPntLVMgmtResetTimer` object with a valid time (0 to 1440 minutes).

Call Stats

The eDVA has completed a call and provides end-of-call statistics.

Event ID: 37

Format:

Call Stats: *LLine*, HW: *model*, SW: *ver*, RTP Tx *tx_pkts*, RTP Rx *rx_pkts*, RTP Lost

pkts_lost, Prov State: *prov_state*, Avg Jtr *jitter_avg*, Max Jtr *jitter_max*, Avg Ltc RTP *ltc_rtp*, Avg Ltc Sig Msg *ltc_sig*, No ACKs *nack*, CMS LOC: *status*

Fields:

The fields are as follows:

- *line*—the line number.
- *model*—the eDVA model number.
- *ver*—the Touchstone firmware version.
- *tx_pkts*—the number of RTP packets sent during the call.
- *rx_pkts*—the number of RTP packets received during the call.
- *pkts_lost*—the number of RTP packets lost during the call, or (**null**) if no packets were lost.
- *prov_state*—the eDVA provisioning state.
- *jitter_avg*—the average jitter during the call.
- *jitter_max*—the maximum jitter during the call.
- *ltc_rtp*—the average latency for RTP packets.
- *nack*—the number of unacknowledged packets.
- *status*—the "loss of comms" status for the eDVA.

Action:

None.

Last NCS Message Received

After a call, the eDVA device provides the last NCS message received during the call.

Event ID: 39

Format:

Last NCS Message Recieved *msg*

Fields:

The fields are as follows:

- *msg*—the text of the NCS message.

Action:

None.

Power Supply Telemetry Alarm - Battery Missing

The Touchstone device either has no battery, or the battery has failed and cannot be detected.

Event ID: 65519

Action:

Replace or install a battery in the device.

Power Supply Telemetry Alarm - Battery Low

The Touchstone device has a battery with a low charge.

Event ID: 65520

Action:

Look for PacketCable "AC Fail" and "AC Restored" messages to determine whether the device is still operating on battery power. Messages from multiple sources indicate a general power outage. If this message is received soon after an "AC Fail" message, the battery may be failing or was not completely charged (for example, soon after a battery test).

Power Supply Telemetry Alarm - Replace Battery

The Touchstone device has a battery that testing indicates has deteriorated to the point where it needs to be replaced.

Event ID: 65521

Action:

Replace the battery as soon as possible.

Voice Line Failure

A voice line has been disabled because one of the following conditions has occurred:

- An In-Service line card has detected a Line Card Protection Fault condition (an over-current protection state). A Line Card Protection Fault occurs when the line card detects foreign voltage between tip and ring, or there is an excessive imbalance in loop current.
- An attempt was made to put an Out-of-Service line, in an overcurrent protection state, into service.

Event ID: 65533

Format:

Voice Line Failure: *line*

Fields:

The fields are as follows:

- *line*—the line that failed.

Action:

Look for Voice Line Protection State Change logs to determine which line is in the fault condition. When you have located the failed line, run line card diagnostics. If the eDVA fails diagnostics, disconnect the house wiring from the eDVA and run diagnostics again. Proceed as follows:

- If the line passes, correct the faulty house wiring.

- If the line fails again, replace the eDVA.

PacketCable Events

The following are PacketCable-related events. PacketCable events use the enterprise number **4491**. All supported PacketCable events are related to the battery and charger system.

Battery Not Low

The E-UE battery has recharged to over 25% of its maximum capacity.

Battery Low

The E-UE battery run time is less than or equal to the value of the **upsConfigLowBattTime** object (which also triggers the **upsAlarmLowBattery** alarm).

Battery Present

A missing battery has been replaced.

If a large number of “Battery Missing” and “Battery Present” messages appear in the logs for a single E-UE, this may indicate a problem with either the battery or the E-UE.

Battery Missing

The E-UE battery has been either removed or is undetectable. If the battery is still installed, it may be defective.

If a large number of “Battery Missing” and “Battery Present” messages appear in the logs for a single E-UE, this may indicate a problem with either the battery or the E-UE.

Battery Good

An E-UE battery that had previously shown a “Replace Battery” state is now good. This may indicate that the battery has been replaced.

Replace Battery

The E-UE battery has deteriorated to about 75% of its off-the-shelf capacity and should be replaced.

AC Restored

AC power has been restored to the indicated E-UE.

If a large number of “AC Restored” and “AC Fail” messages appear in the logs for a single E-UE, this may indicate that the E-UE is connected to a switched outlet or the power connection may be intermittent.

AC Fail

AC power has been removed from the E-UE, and the E-UE is running on battery power.

If a large number of “AC Restored” and “AC Fail” messages appear in the logs for a single E-UE, this may indicate that the E-UE is connected to a switched outlet or the power connection may be intermittent.

Network Failure Recovery

Touchstone firmware can automatically recover from cable cuts or degraded downstream RF conditions. No provisioning or control is necessary to take advantage of recovery features.

Recovery from Extreme Plant Conditions

In a situation where the downstream RF signal degrades to the point where the upstream transmit buffers are full, and the E-UE cannot send packets, an automatic recovery feature:

- resets the DOCSIS layer
- restarts the DSP
- notifies Call Processing to restore dialtone functionality

The result is a cleaner recovery from extreme RF degradation.

Working with Message Trace Logs

Touchstone E-UEs keep logs of:

- CM DHCP messages
- eDVA DHCP messages
- CallP messages
- MGCP
- DSX

Log messages are available only for NCS loads, through SNMP or the troubleshooting pages. Use this procedure to display message trace logs.

Message Capacity

The message log stores up to 250 messages. The buffer size is 25K bytes. SIP loads generate larger messages, reducing the number of messages actually stored.

The DHCP log stores up to 50 CM messages and up to 50 eDVA DHCP messages. Each DHCP buffer is 5K bytes.



Note: The E-UE captures the original DHCP Discover-Offer exchange, and all subsequent Renew exchanges.

The eDVA uses a circular buffer scheme to store messages. When a new message exceeds the available buffer space, the eDVA deletes the oldest messages as needed.

SNMP Overview

Log messages can be up to 4K bytes in length. Since an SNMP string is limited to 256 bytes, retrieving a message through SNMP requires breaking the message up. The objects **arrisMtaDevSignalingLastMsg1** through **arrisMtaDevSignalingLastMsg16** contain the selected message.

Action

Perform the following tasks as needed.

- [Enabling or Disabling Message Tracing](#) 187
- [Viewing Logs Using SNMP](#) 187

Enabling or Disabling Message Tracing

Follow these steps to enable or disable message tracing using SNMP. Trace logs are enabled by default.

1. Specify which message type that you want to enable or disable by setting the **arrisMtaDevLoggingContext** object as follows:

Value	Log Type
0	MGCP (default)
1	CM DHCP
2	eDVA DHCP
3	DSX

2. Enable or disable message tracing for the selected message type by setting the **arrisMtaDevEnableLogging** object: either **disable(0)** or **enable(1)**.

Viewing Logs Using SNMP

Follow these steps to read message logs using SNMP.

1. Specify which message type that you want to view by setting the **arrisMtaDevLoggingContext** object as follows:

Value	Log Type
0	MGCP (default)
1	CM DHCP
2	eDVA DHCP
3	DSX

2. Write a **0** to **arrisMtaDevEnableCallSigLastMsgRpt**. This creates a snapshot of the selected message log and writes the content of the last message to the **arrisMtaDevSignalingLastMsg1** through **arrisMtaDevSignalingLastMsg16** objects.
3. Read the **arrisMtaDevSignalingLastMsg1** through **arrisMtaDevSignalingLastMsg16** objects to view the last log of the selected type. If the message does not require all 16 objects, the unused objects return empty strings.
4. To read older log messages, write a higher value to the **arrisMtaDevEnableCallSigLastMsgRpt** object — for example, a **1** reads the second newest message, **2** reads the message before that, and so on. To refresh the log, write another **0**.

See “Message Capacity” above for the maximum number of each log type that the Telephony Modem stores.

Capturing Signaling Traces



CAUTION

Potential security breach

This feature allows NCS signaling messages to enter the network as clear text. This breaks security as defined in the PacketCable Security specification. IPsec is used to secure the message link between the eDVA and CMS to, among other things, keep the voice keys exchanged between endpoints secure. Unauthorized personnel may potentially be able to monitor a subscriber's voice traffic.

Touchstone firmware can generate a Syslog report that contains a full signaling trace on a Touchstone E-UE. Individual signaling messages may be up to 4000 bytes in length. Since messages of this size would violate the maximum message size limitations of the Syslog server, long messages are broken into blocks of 128 bytes, time stamped, and numbered for reassembly. The eDVA then sends the blocks to the Syslog server IP address defined during normal E-UE provisioning.

Signaling tracing is controlled using the [arrisMtaDevEnableCallpSigTrace](#) MIB object. You can enable or disable message tracing output on an E-UE using an SNMP manager.



Note: Touchstone firmware supports this feature only for capturing NCS signaling traces. Support for capturing SIP signaling traces may be added in a future release.

Controlling Signaling Tracing

Use the [arrisMtaDevEnableCallpSigTrace](#) MIB object (part of the [arrisMtaDevBase](#) MIB) to enable or disable signal tracing. The default value for this object is **disable(0)**.



CAUTION

Potential performance impacts

The number of messages expected as a result of enabling this feature can affect the real-time performance of the E-UE, and may cause network congestion.

Set the object as follows:

- To enable signaling message tracing, set the object to **enable(1)**. Tracing continues until disabled using the MIB object, or the E-UE is reset.
- To disable signaling message tracing, set the MIB object to **disable(0)**.



Note: The signaling trace feature cannot be enabled through either the CM configuration file or the eDVA configuration file.

Interpreting the Signaling Trace Output Data

The following is an example of a single part transmission from the E-UE to a NCS Call server:

```
Oct 21 10: 55: 04 10. 1. 61. 17 Oct 21 10: 55: 03 2005 mta17. dev61 <44> <4115>
<37> <00: 13: 11: 23: 23: E7> <Xmit: (17: 1 of 1) - ' NTFY 8 aal n/2@mta17. dev61
MGCP 1. 0 NCS 1. 0 X: 26002 0: hu ' >
```

The following is an example of a single part Receive from a NCS Call Server to the E-UE:

```
Oct 21 10: 55: 04 10. 1. 61. 17 Oct 21 10: 55: 03 2005 mta17. dev61 <45> <4115>
<38> <00: 13: 11: 23: 23: E7> <Rcv: (19: 1 of 1) - ' 200 8 OK ' >
```

The following output is part of the Syslog header, and appears in both Transmit and Receive trace messages.

Oct 21 10:55:04

Syslog server Date and Time.

10.1.61.17

The IP address of the eDVA that sent the message.

Oct 21 10:55:03 2005

The E-UE generated Data and Time.

mta17.dev61

The FQDN of the E-UE.

<44>

The Syslog message Event Number. It is incremented for each message in the Syslog.

<4115>

The ARRIS Enterprise Number.

The following sections describe the transmit and receive data payloads.

Interpreting the Transmit Data Payload

The following is an example of a single part transmission from the E-UE to a NCS Call server:

```
Oct 21 10: 55: 04 10. 1. 61. 17 Oct 21 10: 55: 03 2005 mta17. dev61 <44> <4115>
<37> <00: 13: 11: 23: 23: E7> <Xmit: (17: 1 of 1) - ' NTFY 8 aal n/2@mta17. dev61
MGCP 1. 0 NCS 1. 0 X: 26002 0: hu ' >
```

The transmit message data can be broken down into two parts, the Header and the signaling data itself. The header indicates that this is a transmitted message from the E-UE; it provides the sequence number of the message, the block number, total number of blocks in this message, and the payload of the signaling message.

The following table shows the data payload that is part of the transmit message trace. The first two parts comprise the header; the next two parts are the actual message.

<37>

An Index number indicating that this is a transmitted message. All transmitted messages are of type 37.

<00:13:11:23:23:E7>

The MAC address of the E-UE that transmitted the message.

Xmit: (17: 1 of 1) -

All transmitted messages start with Xmit. The **17** is a message sequence number. All transmitted messages have a unique sequence number that increases by one for each complete message transmitted. This internal value is a 32-bit unsigned integer value that increments only when tracing is active. Sequence numbers start at zero. The sequence numbers increment only when a message is sent to the Syslog. The “1 of 1” indicates that this is part 1 of a one part message. All parts of the same message have the same sequence number.

'NTFY 8 aaln/2@mta17.dev61

MGCP 1.0 NCS 1.0 X: 26002 O: hu '

The actual signaling message data. The data is surrounded by single quotes. All signaling messages are NULL terminated strings.

Interpreting the Receive Data Payload

The following is an example of a single part Receive from a NCS Call Server to the eDVA:

```
Oct 21 10:55:04 10.1.61.17 Oct 21 10:55:03 2005 mta17.dev61 <45> <4115>
<38> <00:13:11:23:23:E7> <Rcv: (19: 1 of 1) - '200 8 OK ' >
```

The receive message data can be broken down into two parts, the Header and the signaling data. The header indicates that this is a signaling message received by the eDVA; it provides the sequence number of the message, the block number, total number of blocks in this message, and the signaling data received.

The following table shows the data payload that is part of a receive message trace:

<38>

An index number indicating that this is a received signaling message. All received messages are type 38.

<00:13:11:23:23:E7>

The MAC address of the Message Destination.

Rcv: (19: 1 of 1)

All received messages start with Rcv: . The **19** is a message sequence number. All received messages have a unique sequence number that increases by one for every complete message received by the eDVA. This internal value is a 32-bit unsigned integer value and only increments when the tracing is active. Sequence numbers start at zero. All messages have a unique sequence number. The “1 of 1” indicates that this is part 1 of a 1-part message. All parts of the same message have the same sequence number.

'200 8 OK '

The actual received signaling data. The data is surrounded by single quotes.

Signaling Trace Feature Example Output

Below is a small sample output for a typical off-hook and on-hook sequence in NCS. In a real world situation on an actual network Syslog server, there could be messages unrelated to this feature interleaved with these messages:

```
Oct 21 10:55:00 10.1.61.17 Oct 21 10:54:59 2005
mta17.dev61 <37> <4115> <37> <00:13:11:23:23:E7>
<Xmit: (14: 1 of 1) - 'NTFY 7 aal n/2@mta17.dev61 MGCP 1.0 NCS 1.0 X: 25888
0: hd '>
Oct 21 10:55:00 10.1.61.17 Oct 21 10:54:59 2005 mta17.dev61 <38> <4115>
<38> <00:13:11:23:23:E7>
<Rcv: (16: 1 of 1) - '200 7 OK '>
Oct 21 10:55:00 10.1.61.17 Oct 21 10:54:59 2005 mta17.dev61 <39> <4115>
<38> <00:13:11:23:23:E7>
<Rcv: (17: 1 of 1) - 'RQNT 3752 aal n/2@mta17.dev61 MGCP 1.0 NCS 1.0 X:
26001 S: Q: loop R: hf(I), hu(N) '>
Oct 21 10:55:01 10.1.61.17 Oct 21 10:54:59 2005 mta17.dev61 <40> <4115>
<37> <00:13:11:23:23:E7>
<Xmit: (15: 1 of 1) - '200 3752 OK '>
Oct 21 10:55:01 10.1.61.17 Oct 21 10:54:59 2005 mta17.dev61 <41> <4115>
<38> <00:13:11:23:23:E7>
<Rcv: (18: 1 of 2) - 'RQNT 3753 aal n/2@mta17.dev61 MGCP 1.0 NCS 1.0 X:
26002 D: (#|A|D|[2-9]11|0[2-9]11|0T|00|010|11X|[2-9]XXXXXX|[01][2-
9]XXXXXXXXX|0'>
Oct 21 10:55:01 10.1.61.17 Oct 21 10:54:59 2005 mta17.dev61 <42> <4115>
<38> <00:13:11:23:23:E7>
<Rcv: (18: 2 of 2) - '1[1-9]XXXXX|10XXXX|X.#) S: dl R: hf(I,K), hu(N), oc,
of, [0-9*#T](D) '>
Oct 21 10:55:01 10.1.61.17 Oct 21 10:54:59 2005 mta17.dev61 <43> <4115>
<37> <00:13:11:23:23:E7>
<Xmit: (16: 1 of 1) - '200 3753 OK '>
Oct 21 10:55:04 10.1.61.17 Oct 21 10:55:03 2005 mta17.dev61 <44> <4115>
<37> <00:13:11:23:23:E7>
<Xmit: (17: 1 of 1) - 'NTFY 8 aal n/2@mta17.dev61 MGCP 1.0 NCS 1.0 X: 26002
0: hu '>
Oct 21 10:55:04 10.1.61.17 Oct 21 10:55:03 2005 mta17.dev61 <45> <4115>
<38> <00:13:11:23:23:E7>
<Rcv: (19: 1 of 1) - '200 8 OK '>
Oct 21 10:55:04 10.1.61.17 Oct 21 10:55:03 2005 mta17.dev61 <46> <4115>
<38> <00:13:11:23:23:E7>
<Rcv: (20: 1 of 1) - 'RQNT 3788 aal n/2@mta17.dev61 MGCP 1.0 NCS 1.0 X:
26025 S: Q: loop R: hd(N) '>
Oct 21 10:55:04 10.1.61.17 Oct 21 10:55:03 2005 mta17.dev61 <47> <4115>
<37> <00:13:11:23:23:E7>
<Xmit: (18: 1 of 1) - '200 3788 OK'> Oct 21 10:55:33 10.1.61.17 Oct 21
10:55:32 2005 mta17.dev61 <48> <4115> <37> <00:13:11:23:23:E7>
<Xmit: (19: 1 of 1) - 'NTFY 9 aal n/2@mta17.dev61 MGCP 1.0 NCS 1.0 X: 26025
0: hd '>
Oct 21 10:55:33 10.1.61.17 Oct 21 10:55:32 2005 mta17.dev61 <49> <4115>
<38> <00:13:11:23:23:E7>
<Rcv: (21: 1 of 1) - '200 9 OK '> Oct 21 10:55:33 10.1.61.17 Oct 21
10:55:32 2005 mta17.dev61 <50> <4115> <38> <00:13:11:23:23:E7>
<Rcv: (22: 1 of 1) - 'RQNT 3940 aal n/2@mta17.dev61 MGCP 1.0 NCS 1.0 X:
26060 S: Q: loop R: hf(I), hu(N) '>
```

Configuring SNMP Coexistence

The firmware provides several SNMPv3 security-related features. SNMP Coexistence is a feature that allows SNMPv1 and SNMPv2c network management systems to function within the context of SNMPv3 security and view-based MIB access. The NMS can use an SNMPv1 or SNMPv2 community string to access the eMTA's MIB or to receive traps.

Overview

This procedure provides details on adding the necessary MIBs and TLVs to a cable modem configuration file.

To configure the eMTA for coexistence mode, you must create a row entry in the **snmpCommunityTable**, to map the community string to an SNMPv3 security name. You can optionally modify the following tables to provide extended access control.

vacmSecurityToGroupTable

Two row entries, containing group name information. One entry supports SNMPv1 access and the other entry supports SNMPv2 access.

vacmAccessTable

Two row entries, which map the community name, security name, and group name information to an SNMPv3 security name. The DOCSIS-standard default security name for cable modems is **docsisManager**. One entry supports SNMPv1 access and the other entry supports SNMPv2 access.

Enabling SNMP Access

SNMP access is enabled and open to all users until applying any restricted settings in the CM configuration file. The following example fragment can be placed in a CM configuration file to restrict SNMP access:

```
SnmpMib = docsDevNmAccessCommunity.1 "public"
SnmpMib = docsDevNmAccessControl.1 read
SnmpMib = docsDevNmAccessInterfaces.1 hexstr: 40
SnmpMib = docsDevNmAccessStatus.1 createAndGo
SnmpMib = docsDevNmAccessCommunity.2 "private"
SnmpMib = docsDevNmAccessControl.2 readWrite
SnmpMib = docsDevNmAccessInterfaces.2 hexstr: 40
SnmpMib = docsDevNmAccessStatus.2 createAndGo
```

The firmware also allows SNMP access through the RF interface when the CM configuration file does not specify **docsDevNmAccessInterfaces** but includes all other NmAccess entries. For example, the following fragment enables SNMP access through the RF interface:

```
SnmpMib = docsDevNmAccessIp.10 192.168.31.0
SnmpMib = docsDevNmAccessIpMask.10 255.255.255.0
SnmpMib = docsDevNmAccessCommunity.10 "public"
SnmpMib = docsDevNmAccessControl.10 read
SnmpMib = docsDevNmAccessStatus.10 createAndGo
```




Note: AR01.1 allows read-only SNMP access to certain objects from the LAN interfaces before the modem has ranged and registered, in accordance with CM-SP-OSSlv3.0-I08-090121.

Configuration File Notes

Keep the following notes in mind when creating or editing configuration files.

- A MIB object whose type is “StorageType” must always have a value of **volatile**.
- A MIB object whose type is “RowStatus” should have a value of **createAndGo**. The eMTA automatically changes its value to **active** after successfully adding the row.

SNMP Access Mode

The following examples configure SNMP access to the eMTA for SNMPv1v2c coexistence mode. This allows an NMS (i.e. a MIB Browser) to access the eMTA’s MIBs with a simple community string using either SNMPv1 or SNMPv2.

The SNMP requests GET, GET-NEXT, and SET are all supported. The examples use the community name **my_password**.

SNMP Trap Transmission

SNMP trap transmission uses the DOCSIS TLV-38 (SNMPv3 Notification Receiver) configuration file element.

The example configures two trap destinations, each with a different IP address. One destination supports SNMPv1 traps and the other destination supports SNMPv2 traps. The parameters for each trap destination are:

- Trap destination #1:
 - IP Address: 10.1.50.100
 - Trap Type: SNMPv1
- Trap destination #2:
 - IP Address: 10.1.50.80
 - Trap Type: SNMPv2



Note: Change the trap IP address to the IP address of your specific trap server in the configuration file.

This example starts with a basic DOCSIS 1.1 CM configuration file, containing enough information to allow a cable modem to range and register, and then add the coexistence MIB elements to it. If you have a CM configuration file that you are already using, start with that file and add the coexistence elements to it.

snmpCommunityTable Parameters

The following table shows the example row to add to the [snmpCommunityTable](#). The index for this table is an octet string; the example uses the string **comm1** as the index value. You can use a different string if you desire. Italicized values in the table are default values that are created automatically.

The following examples use the ARRIS PacketACE Configuration Editor to create the configuration file, covering only the details needed to add the desired functionality. See the *PacketACE Configuration Tools User's Guide* for more information about using PacketACE.

Object Name	Value	Required?
snmpCommunityName .comm1	my_password	Yes
snmpCommunitySecurityName .comm1	rwAccess	Yes
snmpCommunityContextEngineID .comm1	<i>local snmpEngineID</i>	No
snmpCommunityContextName .comm1	(zero-length)	No
snmpCommunityTransportTag .comm1	(zero-length)	No
snmpCommunityStorageType .comm1	volatile (2)	No
snmpCommunityStatus .comm1	createAndGo (4)	Yes



Note: Avoid adding table index objects to the configuration file; the firmware fills in the index object using the index value supplied with the object name (.comm1 in this example). See ["Adding the snmpCommunityTable"](#) (page 197) for the proper way to add rows.

Object order is not important.

vacmSecurityToGroupTable Parameters

The following table shows the example row to add to the **vacmSecurityToGroupTable**. This table has an index consisting of two objects:

vacmSecurityModel

Corresponds to the SNMP version in use; in this example; it takes the values **1** and **2** to allow support for both SNMPv1 and SNMPv2 requests.

vacmSecurityName

Corresponds to the **snmpCommunitySecurityName** object in the **snmpCommunityTable** (**rwAccess** in this example).

Object Name	Value (row 1)	Value (row 2)
vacmSecurityModel (row index 1)	1 (SNMPv1)	2 (SNMPv2)
vacmSecurityName (row index 2)	rwAccess	rwAccess
vacmGroupName	rwAccess1	rwAccess2
vacmSecurityToGroupStorageType	volatile(2)	volatile(2)
vacmSecurityToGroupStatus	createAndGo(4)	createAndGo(4)

vacmAccessTable Parameters

The following table shows the example row to add to the **vacmAccessTable**. This table has an index consisting of four objects:

vacmGroupName

Corresponds to the **vacmGroupName** object in the **vacmSecurityToGroupTable** (**rwAccess** in our example). CM and MTA group names should be unique.

vacmAccessContentPrefix

An octet string; in this example we use an empty (zero length) string. This is shown as "" in the table below.

vacmAccessSecurityModel

Corresponds to the SNMP version in use; in this example; it takes the values **1** and **2** to allow support for both SNMPv1 and SNMPv2 requests.

vacmAccessSecurityLevel

For SNMP coexistence, use a value of **1** (noAuthnoPriv). This means that the eMTA has no configured SNMPv3 USM security users/keys.

Object Name	Value (row 1)	Value (row 2)
GroupName (row index 1)	rwAccess	rwAccess
ContentPrefix (row index 2)	""	""
SecurityModel (row index 3)	1 (SNMPv1)	2 (SNMPv2)
SecurityLevel (row index 4)	1 (noAuthnoPriv)	1 (noAuthnoPriv)
vacmAccessContextMatch	exact(1)	exact(1)
vacmAccessReadViewName	docsi sManagerVi ew	docsi sManagerVi ew
vacmAccessWriteViewName	docsi sManagerVi ew	docsi sManagerVi ew
vacmAccessNotifyViewName	docsi sManagerVi ew	docsi sManagerVi ew
vacmAccessStorageType	vol atil e(2)	vol atil e(2)
vacmAccessStatus	createAndGo(4)	createAndGo(4)

Action

Perform the following tasks in the order shown.

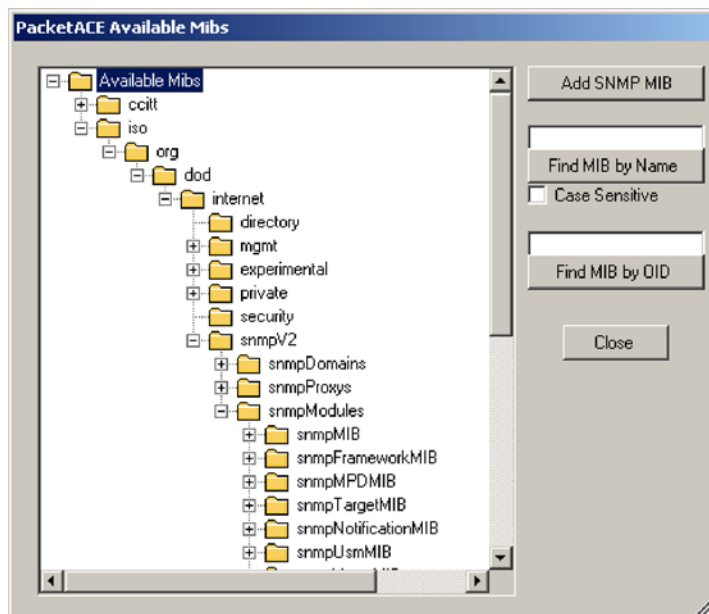
- [1] Adding the snmpCommunityTable..... 197
- [2] Adding the vacmSecurityToGroupTable..... 199
- [3] Adding the vacmAccessTable 201

Adding the snmpCommunityTable

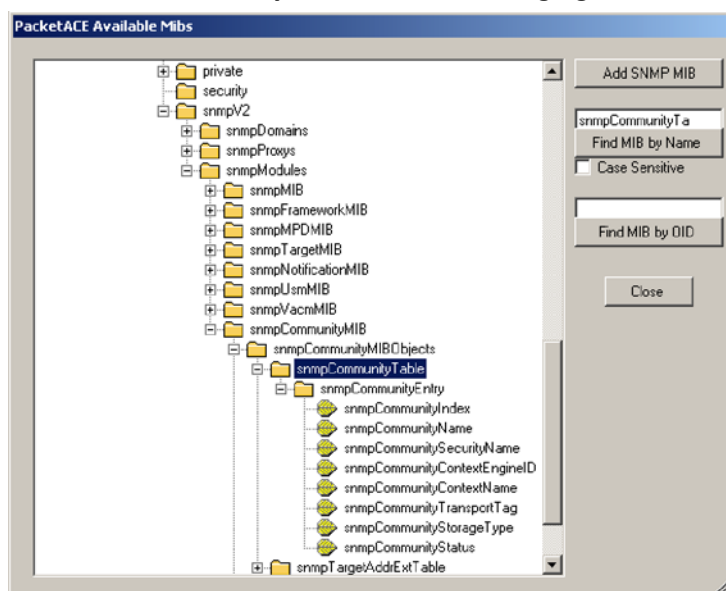
Follow these steps to add coexistence MIB objects to a CM configuration file using PacketACE.

1. Click the “Add SNMP MIB” icon or select **Edit menu** → **Add SNMP MIB**.

The Available MIBs tree appears:



2. Locate the **snmpCommunityTable**. If you are not sure where the table is located in the tree, enter the name in the upper field along the right side of the PacketACE window, then click **Find MIB by Name**. The following figure shows the **snmpCommunityTable**.



3. Double-click on one of the objects listed in the following table.

Object Name	Value
snmpCommunityName	<i>my_password</i>
snmpCommunitySecurityName	rwAccess
snmpCommunityStatus	createAndGo(4)

The Add SNMP MIB window appears:

Add SNMP MIB

MIB: snmpCommunityName

Index, DisplayString:

Value:

OID: 1.3.6.1.6.3.18.1.1.1.2

Variable Type: OCTET STRING

Minimum Value: none

Maximum Value: none

Minimum Size: none

Maximum Size: none

Size: none

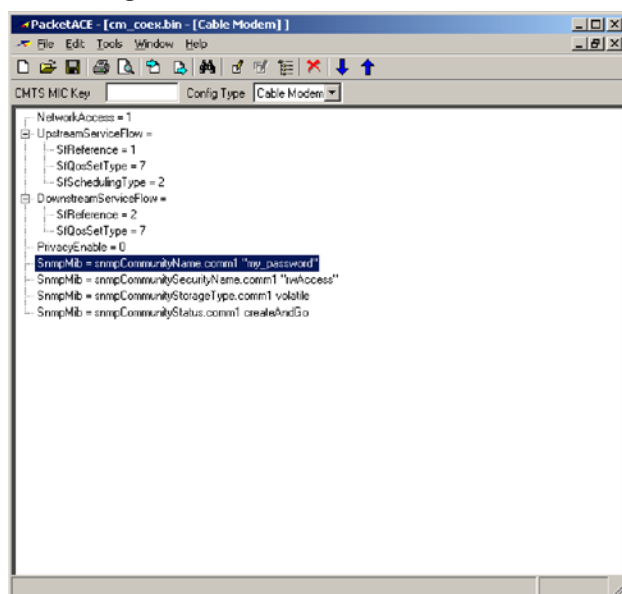
Buttons: Add SNMP MIB, Cancel

4. Enter the index name (**comm1**) in the **Index,DisplayString** field, and the value from the table in step 3 in the **Value** field.

Some of the objects have a fixed set of values; this is indicated by the drop-down menu button at the end of the **Value** field. Click the drop-down to display a list of allowed values, and choose the correct value.

5. Repeat steps 3 and 4 until all values are filled in.

The configuration file should now look similar to the following:





Note: The order of your MIB entries may be different than what is shown above.

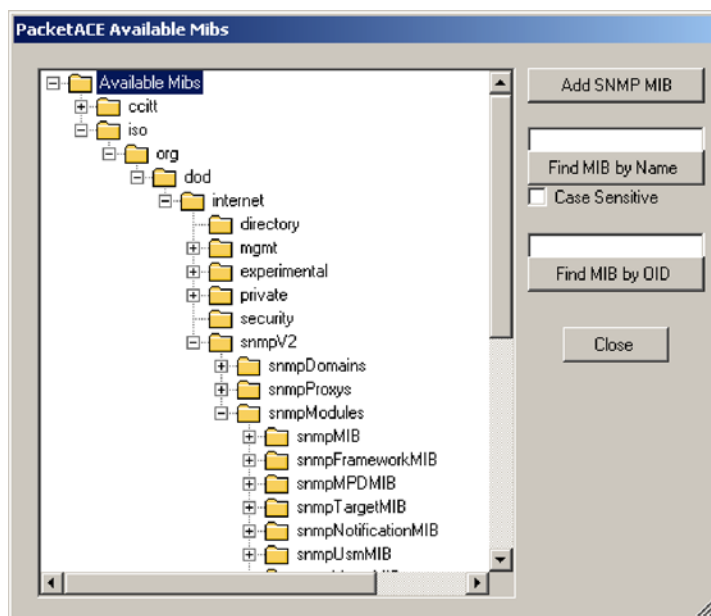
6. Proceed to see "[Adding the vacmSecurityToGroupTable](#) (page 199).

Adding the vacmSecurityToGroupTable

Follow these steps to add coexistence MIB objects to a CM configuration file using PacketACE.

1. Click the "Add SNMP MIB" icon or select **Edit menu** → **Add SNMP MIB**.

The Available MIBs tree appears.



2. Locate the [vacmSecurityToGroupTable](#). If you are not sure where the table is located in the tree, enter the name in the upper field along the right side of the PacketACE window and then click **Find MIB by Name**.
3. Double-click on one of the objects listed in the following table.

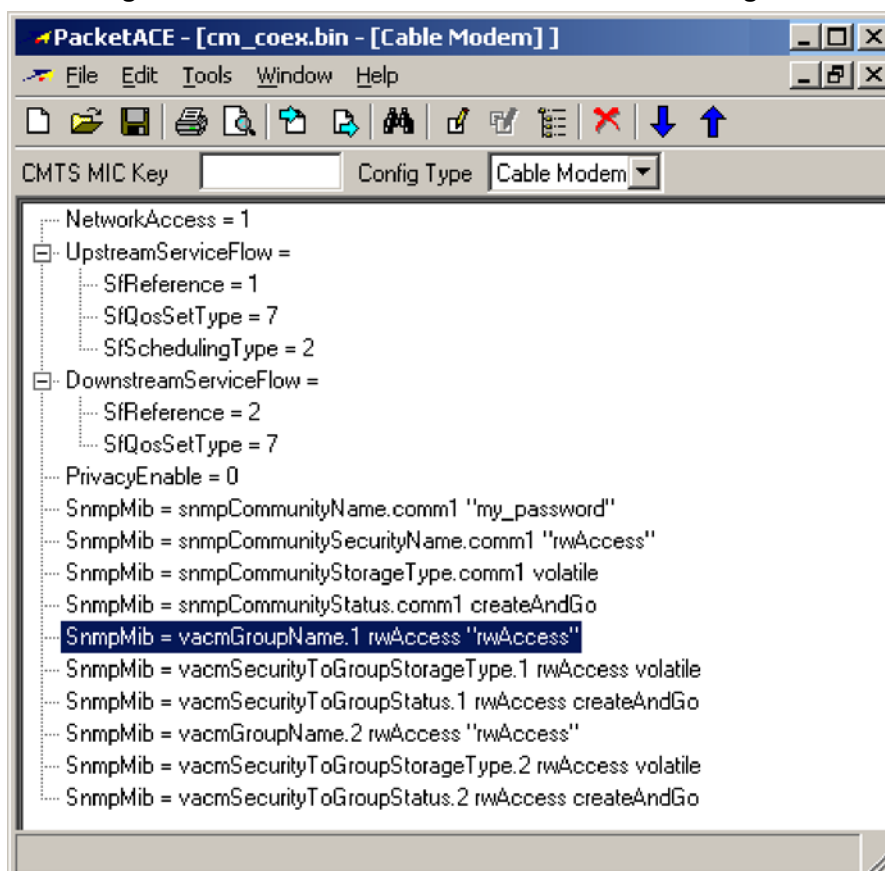
Object Name	Value (row 1)	Value (row 2)
vacmGroupName	rwAccess	rwAccess
vacmSecurityToGroupStorageType	volatile(2)	volatile(2)
vacmSecurityToGroupStatus	createAndGo(4)	createAndGo(4)

The Add SNMP MIB window appears:

Property	Value
OID:	1.3.6.1.6.3.16.1.2.1.3
Variable Type:	SnmpAdminString
Minimum Value:	none
Maximum Value:	none
Minimum Size:	1
Maximum Size:	32
Size:	none

4. Enter the first index (**1** or **2**) in the **Index1,Integer** field, the second index (**rwAccess**) in the **Index2,DisplayString** field, and the value from the table in step 3 in the **Value** field. Some of the objects have a fixed set of values; this is indicated by the drop-down menu button at the end of the **Value** field. Click the drop-down to display a list of allowed values, and choose the correct value.
5. Repeat steps 3 and 4 until all values are filled in.

6. The configuration file should now be similar to the following:



Note: The order of your MIB entries may be different than what is shown above.

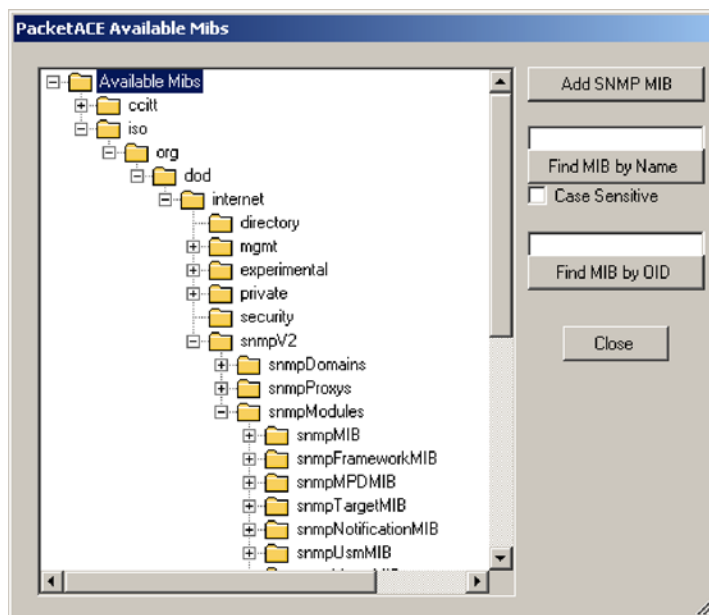
7. Proceed to see "[Adding the vacmAccessTable](#) (page 201).

Adding the vacmAccessTable

Follow these steps to add coexistence MIB objects to a CM configuration file using PacketACE.

1. Click the "Add SNMP MIB" icon or select **Edit menu** → **Add SNMP MIB**.

The Available MIBs tree appears:



2. Locate the **vacmAccessTable**. If you are not sure where the table is located in the tree, enter the name in the upper field along the right side of the PacketACE window, then click **Find MIB by Name**.
3. Double-click on one of the objects listed in the following table.

Object Name	Value (row 1)	Value (row 2)
vacmAccessContextMatch	exact(1)	exact(1)
vacmAccessReadViewName	docsi sManagerVi ew	docsi sManagerVi ew
vacmAccessWriteViewName	docsi sManagerVi ew	docsi sManagerVi ew
vacmAccessNotifyViewName	docsi sManagerVi ew	docsi sManagerVi ew
vacmAccessStorageType	vol atile(2)	vol atile(2)
vacmAccessStatus	createAndGo(4)	createAndGo(4)

The Add SNMP MIB window appears:

Add SNMP MIB

MIB: vacmAccessContextMatch

Index 1, DisplayString: rwAccess

Index 2, DisplayString:

Index 3, INTEGER: 1

Index 4, INTEGER: 1

Value: exact

OID: 1.3.6.1.6.3.16.1.4.1.4

Variable Type: INTEGER

Minimum Value: none

Maximum Value: none

Minimum Size: none

Maximum Size: none

Size: none

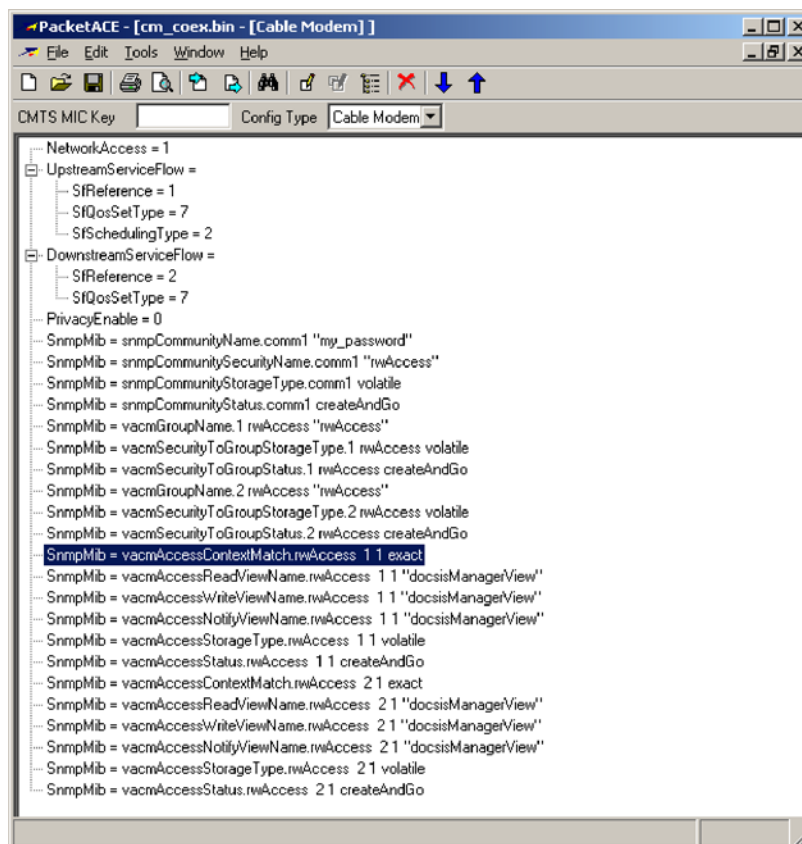
Add SNMP MIB Cancel

4. Enter indexes as follows, and the value from the table in step 3 in the **Value** field.
 - **Index1,DisplayString**— **rwAccess** (or the value of the **vacmGroupName** object from the **vacmSecurityToGroupTable**)
 - **Index2,DisplayString**—leave blank
 - **Index3,Integer**— **1** or **2**, depending on the SNMP level row
 - **Index4,Integer**— **1** (noAuthnoPriv)

Some of the objects have a fixed set of values; this is indicated by the drop-down menu button at the end of the **Value** field. Use the drop-down to display a list of allowed values, and choose the correct value.

5. Repeat steps 3 and 4 until all values are filled in.

The configuration file should now be similar to the following:



Note: MIB entries may be in a different order than what is shown above.

6. Proceed to see "[Configuring Trap Servers](#) (page 204).

Configuring Trap Servers

Use this procedure to configure trap destinations, using TLV-38 SNMPv3 Notification Receiver elements. The following procedure assumes that PacketACE is running and the coexistence MIBs are in the configuration file, but you can add the TLVs to the configuration file before adding the MIBs if you prefer.

Action

Follow these steps to configure trap servers.

1. Click the **"Add TLV parameter"** icon or select **Edit menu → Add TLV**.

The following window appears:

Add TLV

Type: Add TLV

Sub-Type: Cancel

Sub-Sub-Type:

Value:

Variable Type: boolean

Length: 1

2. Select **SNMPv3NotificationReceiver** from the **Type** drop-down menu, then click **Add TLV**.

The **SNMPv3NotificationReceiver** element appears in the main PacketACE window.

Add TLV

Type: Add TLV

Sub-Type: Cancel

Sub-Sub-Type:

Value:

Variable Type: boolean

Length: 1

SNMPv3NotificationReceiver

3. In the main PacketACE window, select the **SNMPv3NotificationReceiver** element, then click the “**Add TLV sub-parameter**” icon or select **Edit menu → Add Sub-TLV**. The following window appears:

Add TLV

Type: Add TLV

Sub-Type: Cancel

Sub-Sub-Type:

Value:

Variable Type: subtypes

Length: variable

SNMPv3NrIpAddress

4. Select **SNMPv3NrIpAddress** from the Sub-Type drop-down menu.
5. Enter the IP address of the trap server (10.1.50.100 in this example) in the Value box.
6. Click **Add TLV**.

PacketACE adds the Trap IP address sub-type to the **SNMPv3NotificationReceiver** element.

7. Repeat steps 4 through 6, adding the SNMPv3NrTrapType sub-parameter and specifying a value of 1: SNMP v1 trap in an SNMP v1 packet.
8. Repeat steps 1 through 7 to create a second SNMPv3Notification-Receiver element with an IP address of 10.1.50.80 and a trap type of 2: SNMP v2c trap in an SNMP v2c packet.

The configuration file should now resemble the following:



9. Save the configuration file and exit PacketACE.

Power Management

Touchstone devices running AR01.1 support enhanced low-power features to maximize battery hold-up time.

When the E-UE loses AC power, it starts the data shutdown timer with a default value of 5 minutes; use the **arrisMtaDevPwrSupplyDataShutdownTime** object to set the desired shutdown time (in seconds). If the timer expires before AC power is restored, the device:

- powers down Ethernet and wireless (if equipped) interfaces. To leave data services in place, set the **arrisMtaDevPwrSupplyEnableDataShutdown** object to **disabled(2)**.
- switches to 1x1 unbonded mode after all lines go on-hook, informing the CMTS using a CM-STATUS message.

Once AC power is restored, the E-UE:

- activates data interfaces (as provisioned)
- attempts to restore its original bonding mode.
- clears codeword counters as a side effect

If the **arrisMtaDevPwrSupplyEnableDataShutdown** is set to **disabled(2)**, the **arrisCmDoc30SetupPowerSaveWifiShutdownOnly** object allows shutting down the wifi interface on Gateway products while leaving the Ethernet interfaces active.

Touchstone Gateway products that support 802.11ac power down LAN-side interfaces immediately on loss of power, regardless of the MIB object settings. This affects the following interfaces:

- MoCA (if equipped)
- Ethernet
- Wi-Fi radios (both 2.4 GHz and 5.0 GHz)
- USB

In this case, the **arrisMtaDevPwrSupplyEnableDataShutdown** and **arrisCmDoc30SetupPowerSaveWifiShutdownOnly** objects control only when the DOCSIS interface switches to 1x1 unbonded operation. For 802.11ac-capable routers, the default time to switch to 1x1 unbonded operation is 30 seconds.

Recovery from Partial Service

The term *partial service* describes the situation where a DOCSIS 3.0 or newer device cannot acquire, or loses after acquisition, one or more of its bonded channels. Section 8.4 of CM-SP-MULPIv3.0-I12-100115 defines the recovery mechanisms to be used under the following scenarios:

When the channel...	The CM signals the CMTS using...
is not acquired during registration	REG-ACK
is not acquired during Dynamic Bonding Change	DBC-RSP
becomes unusable during normal operation	CM-STATUS

Some DOCSIS 3.0 CMTSs currently do not support all three partial service signaling methods. Use the information in this procedure to configure the Telephony Modem to work around the limitations of the CMTS, if necessary.

Consult the release notes for your CMTS firmware or software to determine whether one of the following workarounds are needed.

Action

Perform the following tasks as needed.

- [Identifying Partial Service Issues](#) 208
- [If the CMTS does not Support CM-STATUS](#) 208
- [If the CMTS does not Support REG-ACK](#) 209

Identifying Partial Service Issues

To identify partial service issues using an SNMP browser, read the **arrisCmDoc30ProvisionedChannelIDs** object. This object displays the channel IDs of the provisioned downstreams and upstreams. An example display is:

DS: 8, 5, 6, (7) -- US: 5, 6, 7, 8

This example shows that the downstream channel DCID 7 is down (enclosed in parentheses) and the modem thus has partial service.

If the CMTS does not Support CM-STATUS

If the CMTS supports the REG-ACK Partial Service Confirmation Code, but not CM-STATUS message reporting, follow these steps.

1. Set the **arrisCmDoc30SetupSecDsLossReinitEnable** object to **enable(1)**, using an SNMP manager or by changing the configuration file.
2. If you made the change in the configuration file, reset the Telephony Modem so the change takes effect.

If QAM or FEC lock is lost on a secondary downstream for 45 seconds, the Telephony Modem re-initializes the downstream MAC; this allows a REG-ACK to inform the CMTS which downstreams are available.

3. When the secondary downstream becomes available, manually reset the CM to restore full service.

If the CMTS does not Support REG-ACK

If the CMTS supports bonded channels, but does not support the REG-ACK Partial Service Confirmation Code, follow these steps.

1. Set the **arrisCmDoc30SetupPartServiceFallback20** object to **enable(1)**, using an SNMP manager or by changing the configuration file.
2. If you made the change in the configuration file, reset the Telephony Modem so the change takes effect.

If the Telephony Modem cannot acquire a secondary downstream, it re-registers in DOCSIS 2.0 (non-bonded) mode. Every 15 minutes, the Telephony Modem re-initializes the primary downstream DOCSIS MAC until it can acquire all downstream channels. No further operator intervention is required.

ARRIS DOCSIS 3.0 MIB

The **arrisCmDoc30** MIB provides information, access control, and DHCP settings for Touchstone products supporting DOCSIS 3.0. Objects in this MIB appear in the following groups:

- **arrisCmDoc30Base**
- **arrisCmDoc30Access**
- **arrisCmDoc30Setup**
- **arrisCmDoc30Dhcp**

arrisCmDoc30Base

Miscellaneous information and control objects:

arrisCmDoc30ResetFactoryDefaults

Set to **1** to reset the E-UE to factory default settings.

arrisCmDoc30FwImageName

A string containing the firmware image name.

arrisCmDoc30FwImageBuildTime

A string containing the firmware build date and time. This may be useful when calling ARRIS Technical Support.

arrisCmDoc30BondingMode

The registration mode of the Telephony Modem, how many downstreams and upstreams it uses, and whether (for DOCSIS 3.1 modems) how many OFDM channels are in use. Example: **DOCSIS3.1 32x4 (2 OFDM)**

arrisCmDoc30ResetAccessTime

Set to **true**(1) to reset access start times in non-volatile memory.

arrisCmDoc30ProvisionedChannelIDs

Displays the channel IDs of the provisioned upstreams and downstreams. Downstreams that are not currently locked are enclosed in parentheses, indicating that the modem has partial service.

arrisCmDoc30BaseReportDuplex

Displays the duplex status of the Ethernet port; one of: **Ful l**, **Hal f**, or **Unavai l abl e**.

arrisCmDoc30Access

These objects control access to the CLI and web pages.

arrisCmDoc30AccessTelnetPassword

A string containing the password used to enable Telnet.

arrisCmDoc30AccessClientSeed

A string containing the seed used to generate the Password of the Day. If you change this value, you must also change the value of the seed used in the PacketACE Password of the Day generator. Clear this value to use the default seed.

arrisCmDoc30AccessHttpLan

Controls access to the web pages from the LAN (Ethernet and USB) interfaces.

arrisCmDoc30AccessHttpWan

Controls access to the web pages from the WAN (cable) interface.

arrisCmDoc30AccessHttpTimeout

The time, in minutes, that the Advanced web pages are accessible before the Telephony Modem requires re-entry of the Password of the Day. Use **0** to disable the timeout.

arrisCmDoc30CLITimeout

The time, in minutes, that a Telnet or SSH session can be idle before the Telephony Modem terminates the session. Valid range: **1** to **65535** minutes, or **0** to disable timeout.

arrisCmDoc30AccessHttpPwCtrl

Controls which web pages are password protected:

- **none**(0): no pages are protected
- **advanced**(1): advanced pages are protected (default)
- **al l** (2): all pages are protected

arrisCmDoc30AccessSSHEnable

Enables or disables SSH access. To enable SSH, this object must be enabled, and the **arrisCmDoc30AccessTelnetPassword** object must also be set.

arrisCmDoc30Setup

These objects control cable modem features in Touchstone products.

arrisCmDoc30SetupDSTPolicy

Sets the Daylight Savings Time (DST) policy. This object contains a string, defining the starting and ending dates and times for DST. The format of the string is as follows:

start=*month/day/weekday/hour*;**end**=*month/day/weekday/hour*

where...	is...
month	the month: 1 for January, to 12 for December.
weekday	The day of the week that DST begins or ends: 1 for Monday, to 7 for Sunday, or 0 to ignore the weekday and use the exact date. If not zero, DST begins or ends on the specified weekday after the date if the date is positive, or before the date if negative.
day	The day: -31 to -1 to count backwards from the end of the month, 1 to 31 to count forward from the beginning of the month.
hour	The hour at which DST begins or ends: 00 to 23 .

Example:

start=3/8/7/02; end=11/1/7/02

Implements the U.S. DST policy in effect since March 2007: DST begins at 2 a.m. on the second Sunday in March and ends at 2 a.m. on the first Sunday in November.

arrisCmDoc30SetupTODTimeOffset

Sets the CM ToD time offset from GMT, in seconds. Valid range: **-43200** (-12 hours) to **46800** (+13 hours). This value may be overwritten when the TOD offset option is received through DHCP.

arrisCmDoc30SetupTODSyncTimeout

The interval, in hours, between ToD sync operations. Valid range: **0** (default, disable sync completely) to **4320**.

Setting this object restarts the timer.

arrisCmDoc30SetupIgnoreMddSymbolClockIndicator

When set to **enable(1)**, the CM ignores the Symbol Clock Locking Indicator TLV in the MDD message and always uses asynchronous timing mode. In this mode, the cable modem does not lock to the downstream symbol clock on its Primary Downstream Channel. It acquires the synchronization timebase for upstream burst timing from the SYNC messages.

When set to **disable(0)**, the CM follows the timing mode specified by the Symbol Clock Indicator TLV in the MDD message.

arrisCmDoc30SetupTftpBlkSize

Controls the TFTP Blocksize Option, defined in RFC 2348, for firmware downloads. Valid settings:

- 0: use the default block size of 1428 for IPv6 connections or 1448 for IPv4 connections.
- 64–65464: the TFTP block size, in octets.



Note: Specifying extremely high or low values may affect TFTP performance.

arrisCmDoc30SetupTftpTimeout

Controls the TFTP Timeout Option, defined in RFC 2349, for firmware downloads. Valid settings:

- 0 (default): disable this option.
- 1–255: when smart download mode ([arrisCmDoc30EsafeSmartDownloadMode](#)) is enabled, the TFTP client uses the specified timeout value.

arrisCmDoc30SetupRCPBypass

Controls the RCP check bypass feature. Valid settings:

- **disable**(0): disables RCP check bypass, and overrides any setting of this object in the CM configuration file. If set in an SNMP browser, the value is persistent.
- **enable**(1): enables RCP check bypass, and overrides any setting of this object in the CM configuration file. If set in an SNMP browser, the value is persistent.
- **default**(2): disables RCP check bypass, unless specified otherwise in the CM configuration file. This value is not persistent.

arrisCmDoc30SetupTimeoffsetWrapper

Controls the Time Offset Wrapper feature. Valid settings:

- **disable**(0): disables Time Offset Wrapper, and overrides any setting of this object in the CM configuration file. If set in an SNMP browser, the value is persistent.
- **enable**(1): enables Time Offset Wrapper, and overrides any setting of this object in the CM configuration file. If set in an SNMP browser, the value is persistent.
- **default**(2), disables Time Offset Wrapper, unless specified otherwise in the CM configuration file. This value is not persistent.

arrisCmDoc30DiplexerControl

Displays or sets the diplexer. Valid settings:

- **band0**(0)
- **band1**(1)

Use the [arrisCmDoc30DiplexerFrequencyRanges](#) object to view the frequency ranges for each diplexer.



Note: You must reset the device to allow the diplexer setting to take effect.

arrisCmDoc30DiplexerFrequencyRanges

Displays the upstream and downstream frequency ranges for each diplexer.

Example: Band0: 5- 85MHz/108- 1002MHz; Band1: 5- 42MHz/108- 1002MHz

arrisCmDoc30SetupExtendedUpstreamTransmitPowerEnable

Set to **enable**(1) to report extended upstream transmit power capabilities in the REG-REQ-MP message. Supported only on Model 16 and Model 24 products.

arrisCmDoc30Dhcp

These objects provide DHCP status information. There are four groups under this MIB:

- **arrisCmDoc30DhcpLeaseParameters**
- **arrisCmDoc30DhcpSvrParameters**
- **arrisCmDoc30DhcpCmParameters**
- **arrisCmDoc30DhcpMtaParameters**

arrisCmDoc30DhcpLeaseParameters Objects

These objects provide information about the current DHCP lease.

arrisMtaDoc30DhcpOfferedLeaseTime

The offered lease time, in seconds.

arrisCmDoc30DhcpTimeUntilRenew

The time, in seconds, before the eDVA begins a RENEW exchange.

arrisCmDoc30DhcpTimeUntilRebind

The time, in seconds, before the eDVA begins a REBIND exchange.

arrisCmDoc30DhcpLeaseTimeRemaining

The remaining lease time, in seconds.

arrisCmDoc30DhcpSvrParameters Objects

These objects provide information about the DHCP server. All these objects are read-only.

arrisCmDoc30DhcpState

The current CM DHCP state; one of:

- **init-selecting**(0)
- **requesting**(1)
- **bound**(2)
- **renewing**(3)
- **rebinding**(4)
- **init-reboot**(5)
- **renew-requested**(6)
- **released**(7)

arrisCmDoc30DhcpPrimaryDhcpServerIpAddr

The primary DHCP server address.

arrisCmDoc30DhcpTftpSvrIpAddr

The current TFTP server IP address in use.

arrisCmDoc30DhcpTimeSvrIpAddr

The current Time server IP address in use.

arrisCmDoc30DhcpCmTimeOffset

The current time offset being used by the CM.

arrisCmDoc30DhcpPrimaryTeleDhcpSvr

The primary eDVA DHCP server address.

arrisCmDoc30DhcpSecondaryTeleDhcpSvr

The secondary eDVA DHCP server address.

arrisCmDoc30DhcpCmParameters Objects

These objects provide information about the current CM DHCP lease. All these objects are read-only.

arrisCmDoc30DhcpCmIpAddrType

The type (IPv4 or IPv6) of the currently leased IP address.

arrisCmDoc30DhcpCmIpAddr

The currently leased IP address.

arrisCmDoc30DhcpCmSubNetMask

The current IP subnet mask in use.

arrisCmDoc30DhcpCmGatewayIpAddr

The current IP gateway address in use.

arrisCmDoc30DhcpCmConfigFile

The CM configuration file to be retrieved.

arrisCmDoc30DhcpMtaParameters Objects

These objects provide information and control over eDVA DHCP operation.

arrisCmDoc30DhcpMtaOpt60Override

Enables or disables SIP advertisement in eDVA DHCP Option 60. The default is disabled.

This object can be set only using the CM configuration file.

arrisCmDoc30DhcpExtended

These objects provide extended DHCP information. All these objects are read-only.

arrisCmDoc30DhcpExtendedProvisionedMode

Displays the CM provisioned mode:

- **i p v 4- o n l y**(0)
- **i p v 6- o n l y**(1)
- **a l t e r n a t e- p r o v- m o d e**(2)
- **d u a l - p r o v- m o d e**(3)

arrisCmDoc30DhcpExtendedPreferredMode

Displays the CM preferred IP mode: **i pv4(0)** or **i pv6(1)**.

arrisCmDoc30DhcpExtendedActiveMode

Displays the CM active IP mode: **i pv4(0)** or **i pv6(1)**.

arrisCmDoc30DhcpExtendedLeaseParametersTable Objects

This table provides DHCP lease information.

arrisCmDoc30DhcpExtendedLeaseParametersType

The DHCP type: **dhcipv4(0)** or **dhcipv6(1)**.

arrisCmDoc30DhcpExtendedOfferedLeaseTime

The offered IP lease time, in seconds.

arrisCmDoc30DhcpExtendedTimeUntilRenew

The current time remaining, in seconds, before the eDVA starts the lease renewal process.

arrisCmDoc30DhcpExtendedTimeUntilRebind

The current time remaining, in seconds, before the CM starts the lease rebinding process.

arrisCmDoc30DhcpExtendedLeaseTimeRemaining

The remaining IP lease time, in seconds.

arrisCmDoc30DhcpExtendedRenewLease

Set to **apply(1)** to renew the WAN(0) DHCP lease.

arrisCmDoc30DhcpExtendedSvrParametersTable

This table provides extended information about the DHCP server.

arrisCmDoc30DhcpExtendedSvrParametersType

The DHCP type: **dhcipv4(0)** or **dhcipv6(1)**.

arrisCmDoc30DhcpExtendedState

The current DHCP state of the CM:

- **i n i t - s e l e c t i n g(0)**
- **r e q u e s t i n g(1)**
- **b o u n d(2)**
- **r e n e w i n g(3)**
- **r e b i n d i n g(4)**
- **i n i t - r e b o o t(5)**
- **r e n e w - r e q u e s t e d(6)**
- **r e l e a s e d(7)**
- **d h c p 6 c - i n i t(8)**
- **d h c p 6 c - s o l i c i t(9)**

- **dhcp6c- i nforeq**(10)
- **dhcp6c- request**(11)
- **dhcp6c- renew**(12)
- **dhcp6c- rebi nd**(13)
- **dhcp6c- rel ease**(14)
- **dhcp6c- decl i ne**(15)
- **dhcp6c- confi rm**(16)
- **dhcp6c- i dl e**(17)

arrisCmDoc30DhcpExtendedPrimaryDhcpServerIpAddr

The primary DHCP server address.

arrisCmDoc30DhcpExtendedTftpSvrIpAddr

The current TFTP server IP address in use.

arrisCmDoc30DhcpExtendedTimeSvrIpAddr

The current time server IP address in use.

arrisCmDoc30DhcpExtendedCmTimeOffset

The current time offset used by the CM.

arrisCmDoc30DhcpExtendedPrimaryTeleDhcpSvr

The primary eDVA DHCP server address.

arrisCmDoc30DhcpExtendedSecondaryTeleDhcpSvr

The secondary eDVA DHCP server address.

arrisCmDoc30DhcpExtendedSrvDUIDV6

(read-only) The server DUID.

arrisCmDoc30DhcpExtendedCmParametersTable Objects

This table provides extended DHCP information for the CM.

arrisCmDoc30DhcpExtendedCmParametersType

The DHCP type: **dhcpv4**(0) or **dhcpv6**(1).

arrisCmDoc30DhcpExtendedCmIpAddr

The currently leased IP address.

arrisCmDoc30DhcpExtendedCmSubNetMask

The current IP subnet mask.

arrisCmDoc30DhcpExtendedCmPrefix

The current IP Prefix.

arrisCmDoc30DhcpExtendedCmGatewayIpAddr

The current IP gateway address.

arrisCmDoc30DhcpExtendedCmConfigFile

The CM configuration file to be retrieved.

arrisCmDoc30DhcpExtendedCmDUIDV6

The CM DUID.

arrisCmDoc30ResetReasonLog

These objects contain the reset reason log. All these objects are read-only.

arrisCmDoc30LastHwResetReason

The last reset reason retrieved from the processor hardware. This reason is used to determine if hardware or firmware caused the reset.

arrisCmDoc30ResetReasonLogTable Objects

This table contains the last ten reset reasons. The valid index range is **1** to **10**.

arrisCmDoc30ResetReasonLogText

The reported reset reason log text.

HD Audio MIB Objects

The following MIB objects are under **arrisMtaDevOperationalSetup**, and provision and control HD voice:

arrisMtaDevWBSLIC

Set to **enable**(1) to enable wideband audio support in the eDVA hardware. The default value is **disable**(0). The setting is persistent across reboots.

- Typically, you should add the desired setting to the CM configuration file. If the object value in the CM configuration file is different from the current setting, the device automatically restarts.
- If you change this object using SNMP, you must reboot the device for the new setting to take effect.

arrisMtaDevHDAudioEnable

(TS8.1 and newer) Set to **enable**(1) (the default) to advertise and allow negotiation of the G722 CODEC.

arrisMtaDevHDAudioEndPntEnable

Enables individual line support of HD voice. This object is indexed by line number, starting with 1.

arrisMtaDevHDAudioDefaultPayloadType

Specifies static or dynamic payload type in the SDP. The default value, **static**(0), offers static payload type 9 and does not include an RTPmap in the SDP. If set to **dynamic**(1), the eDVA offers a payload range of 96 to 127, and includes an RTPmap in the SDP.



Note: this object only affects which payloads are offered in the initial SDP. CODEC negotiation determines the actual payload type and CODEC used, as with narrowband CODECs.

arrisMtaDevHDAudioG722SampleRate

Sets the default parameter in the RTPmap for G.722 CODECs. The default, **rate8000(0)**, sets a sampling rate of 8000. Set this object to **rate16000(1)** to set a sampling rate of 16000.



Note: When a G.722 CODEC is in use, and tel phone- event is negotiated, the sampling rate is 16000 regardless of this object's setting.

DOCSIS 3.0 MIB Object Mapping

The following table shows how to map certain ARRIS DOCSIS 2.0 MIB objects to their DOCSIS 3.0 counterparts.

DOCSIS 2.0 Object/OID	DOCSIS 3.0 Object/OID
arrisCmProdResetToFactoryDefaults	arrisCmDoc30ResetFactoryDefaults
1.3.6.1.4.1.4115.1.3.2.2.15	1.3.6.1.4.1.4115.1.3.4.1.1.6
arrisCmDevSwImageName	arrisCmDoc30FwImageName
1.3.6.1.4.1.4115.1.3.1.1.1.2	1.3.6.1.4.1.4115.1.3.4.1.1.7
arrisCmDevSwImageBuildTime	arrisCmDoc30FwImageBuildTime
1.3.6.1.4.1.4115.1.3.1.1.1.3	1.3.6.1.4.1.4115.1.3.4.1.1.8
arrisCmProdAccessPWD	arrisCmDoc30AccessTelnetPassword
1.3.6.1.4.1.4115.1.3.2.2.2	1.3.6.1.4.1.4115.1.3.4.1.2.1
arrisCmDevTelnetEnable	arrisCmDoc30AccessTelnetEnable
1.3.6.1.4.1.4115.1.3.1.1.2.3.23	1.3.6.1.4.1.4115.1.3.4.1.2.2
arrisCmDevHttpClientSeed	arrisCmDoc30AccessClientSeed
1.3.6.1.4.1.4115.1.3.1.1.2.3.5.5	1.3.6.1.4.1.4115.1.3.4.1.2.3
arrisCmDevHttpLanAccess	arrisCmDoc30AccessHttpLan
1.3.6.1.4.1.4115.1.3.1.1.2.3.5.3	1.3.6.1.4.1.4115.1.3.4.1.2.5
arrisCmDevHttpWanAccess	arrisCmDoc30AccessHttpWan
1.3.6.1.4.1.4115.1.3.1.1.2.3.5.4	1.3.6.1.4.1.4115.1.3.4.1.2.6
arrisCmDevSwCustomerLoadId	arrisCmDoc30SwCustomerLoadId
1.3.6.1.4.1.4115.1.3.3.1.4.1.1.2	1.3.6.1.4.1.4115.1.3.4.1.5.1.1.2
arrisCmDevSwHwModel	arrisCmDoc30SwHwModel
1.3.6.1.4.1.4115.1.3.3.1.4.1.1.3	1.3.6.1.4.1.4115.1.3.4.1.5.1.1.3
arrisCmDevSwHwRev	arrisCmDoc30SwHwRev

DOCSIS 2.0 Object/OID	DOCSIS 3.0 Object/OID
1.3.6.1.4.1.4115.1.3.3.1.4.1.1.4	1.3.6.1.4.1.4115.1.3.4.1.5.1.1.4
arrisCmDevSwFilename	arrisCmDoc30SwFilename
1.3.6.1.4.1.4115.1.3.3.1.4.1.1.6	1.3.6.1.4.1.4115.1.3.4.1.5.1.1.5
arrisCmDevSwServerAddressType	arrisCmDoc30SwServerAddressType
1.3.6.1.4.1.4115.1.3.3.1.4.1.1.7	1.3.6.1.4.1.4115.1.3.4.1.5.1.1.6
arrisCmDevSwServerAddress	arrisCmDoc30SwServerAddress
1.3.6.1.4.1.4115.1.3.3.1.4.1.1.8	1.3.6.1.4.1.4115.1.3.4.1.5.1.1.7

Supported eDVA MIB Objects

AR01.1 supports the following ARRIS eDVA MIB objects.

PACKETPORT-MIB Objects

The following objects are supported:

ppCfgMtaCountryTemplate

Specifies the country template for ring cadences and signaling tones.

ppCfgMtaCallpFeatureSwitch

The CallP Feature Switch. See *CallP Feature Switch* (page 40) for details.

ppCfgRfc2833DigitPayloadType

When bit 0x01000000 of the CallP Feature Switch is set, this object specifies the payload type used to send DTMF events. Valid range: **97** to **127**. Default: **101**.

ppCfgMtaFeatureSwitch

The MTA Feature Switch.

ppCfgMtaCallpFeatureSwitch2

The secondary CallP Feature Switch. See *CallP Feature Switch* (page 40) for details.

ppCfgPortTable Objects

The following objects are supported.

ppCfgPortLoopCurrent

Sets the loop current for the specified line: **normal** (1) or **high**(2).

ppCfgPortLocUserIndication

The tone applied during Loss of Communication (LoC): **silence**(0) or **reorderTone**(1) (default).

ppCfgPortT38MaxDatagram

The maximum datagram size for incoming T.38 packets. Valid range: **160** (default) to **65535**. If this object is set to a value higher than the default, the Call Agent must allocate more bandwidth accordingly.

ppSurvPortTable Objects

The following objects are supported.

ppSurvPortMaintState

The maintenance state of the line.

ppSurvPortLcDiagRequest

Set to **true**(2) to begin linecard diagnostics.

ppSurvPortLcDiagLastResult

The last result of linecard diagnostics performed on the line.

ARRIS-MTA-MIB (non-battery)

The following MIB objects from the ARRIS-MTA-MIB are supported. Battery-related objects are listed below.

arrisMtaDevControl Objects

The following **arrisMtaDevControl** objects are supported:

arrisMtaDevResetCallStats

Set this object to **true**(1) to reset the following objects to their default values:

- **arrisMtaDevRtpTxPktsTotal**
- **arrisMtaDevRtpRxPktsTotal**
- **arrisMtaDevRtpPercentPktsLostTotal**
- **arrisMtaDevSignalingAvgLatency**
- **arrisMtaDevSignalingTxSuccessfulMsgCnt**
- **arrisMtaDevSignalingRxSuccessfulMsgCnt**
- **arrisMtaDevSignalingTxNAKCnt**
- **arrisMtaDevSignalingRxNAKCnt**
- **arrisMtaDevSignalingRxNoACKCnt**

Setting this object to **false**(2) does nothing. Reading this object always returns **false**(2).

arrisMtaDevEnableCallpSigTrace

Controls CallP signaling message tracing in the Syslog. Take care when setting this object,

as excessive messaging could adversely affect performance. The default value is **disable(0)**.

arrisMtaDevEnableCallStatsSyslogRpt

Enables end-of-call statistics reporting, and CallP signaling last message reporting, to the Syslog.

When set to **enable(1)**, end-of-call statistics are reported in the Syslog. If the **arrisMtaDevEnableCallSigLastMsgRpt** object is enabled, then the last 4K of signaling messages is also reported in the Syslog.

When set to **disable(0)** (the default), end-of-call statistics and the CallP last signaling messages are not reported in the Syslog.

arrisMtaDevSwDnldNoSvcImpact

Enables or disables the software download service impact feature. When set to **enable(1)** (the default), the eDVA accepts the load, but does not apply the load until all lines have been idle for at least 30 seconds after the load has been accepted.

arrisMtaDevEnableCallSigLastMsgRpt

Enables or disables reporting of the CallP signaling “last message” to the MIB objects **arrisMtaDevSignalingLastMsg1** through **arrisMtaDevSignalingLastMsg16**. Together the sixteen objects contain a signaling message up to 4000 bytes long. Each object contains a 255-byte segment of the message. If the message does not require all sixteen MIBs, then the empty objects display “Buffer is empty.” The default value is **disable(0)**.

arrisMtaDevNsadSwDnldStatus

(read-only) Displays the current firmware download status:

download-Idle(0)

Indicates that the firmware download has completed. This value is also set at startup.

download-Acceptance-In-Progress(1)

The unit is currently downloading the firmware in the background.

download-Application-Pending(2)

Indicates that the load has been downloaded and accepted but is waiting to be applied.

This MIB object is only valid if the **arrisMtaDevSwDnldNoSvcImpact** object is set to **enable(1)**.

arrisMtaDevRestoreNvmFactoryDefault

Set this object to **true(1)** to reset the NVM to default values.

arrisMtaDevTrace Objects

These objects control and display message trace results.

arrisMtaDevRtpTxPktsTotal

(read-only) The total number of RTP packets sent from the eDVA since it was last started

up or reset. This value represents the total number of packets sent for all endpoints combined. It is calculated from values obtained from the signaling end of call statistics. Set the **arrisMtaDevResetCallStats** object to **true(1)** to clear this counter.

arrisMtaDevRtpRxPktsTotal

(read-only) The total number of RTP packets received by the eDVA since it was last started up or reset. This value represents the total number of packets received for all endpoints combined. It is calculated from values obtained from the signaling end of call statistics. Set the **arrisMtaDevResetCallStats** object to **true(1)** to clear this counter.

arrisMtaDevRtpPercentPktsLostTotal

(read-only) The percentage of RTP packets lost since the eDVA was last started up or reset. This value represents the total number of packets lost for all endpoints combined. It is calculated from values obtained from the signaling end of call statistics. Set the **arrisMtaDevResetCallStats** object to **true(1)** to clear this counter.

The value of this object is expressed in increments of 1/100 percent. For example, a value of **1745** means that 17.45% of the packets were lost.

arrisMtaDevRtpPktsLostTotal

(read-only) The number of RTP packets lost since the MTA was last started up or reset. This value represents the total number of packets lost for all endpoints combined. It is calculated from values obtained from the signaling end of call statistics. Set the **arrisMtaDevResetCallStats** object to **true(1)** to clear this counter.

arrisMtaDevLastCallStartTime

(read-only) The last call start time from the eDVA.

arrisMtaDevLastCallEndTime

(read-only) The last call end time from the eDVA.

arrisMtaDevSignalingAvgLatency

(read-only) The average latency or delay, in milliseconds, for responses to signaling messages. It is calculated from values obtained from the signaling end of call statistics. Set the **arrisMtaDevResetCallStats** object to **true(1)** to clear this counter.

arrisMtaDevSignalingTxSuccessfulMsgCnt

(read-only) The total number of successful signaling messages sent from the eDVA. Set the **arrisMtaDevResetCallStats** object to **true(1)** to clear this counter.

arrisMtaDevSignalingRxSuccessfulMsgCnt

(read-only) The total number of successful signaling messages received by the MTA. Set the **arrisMtaDevResetCallStats** object to **true(1)** to clear this counter.

arrisMtaDevSignalingTxNAKCnt

(read-only) The total number of negative acknowledgement signaling messages (NAKmessages) sent from the MTA. Set the **arrisMtaDevResetCallStats** object to **true(1)** to clear this counter.

arrisMtaDevSignalingRxNAKCnt

(read-only) The total number of negative acknowledgement signaling messages

(NAKmessages) received by the MTA. Set the **arrisMtaDevResetCallStats** object to **true**(1) to clear this counter.

arrisMtaDevProvState

(read-only) The current provisioning state of the MTA:

- **dhcpBound**(1)
- **dnsReqProvSvrIP**(2)
- **kdcHostNameDnsReq**(3)
- **kdcHostNameDnsRply**(4)
- **kdcIPDnsReq**(5)
- **kdcIPDnsRply**(6)
- **asReqSent**(7)
- **asRplyRcvd**(8)
- **tgsReqSent**(9)
- **tgsRplyRcvd**(10)
- **apReqSent**(11)
- **apRplyRcvd**(12)
- **enrolmentInform**(13)
- **cfgUrlSet**(14)
- **dnsReqTftpSvrIP**(15)
- **cfgFileReq**(16)
- **rcvCfgFile**(17)
- **syslogMsgProvComplete**(18)
- **statusInform**(19)
- **provcomplete**(20)

arrisMtaDevSWUpgradeStatus

(read-only) The current software upgrade status of the device:

- **inProgress**(1)
- **completeFromProvisioning**(2)
- **completeFromMgt**(3)
- **failed**(4)
- **other**(5)

arrisMtaDevSignalingRxNoACKCnt

(read-only) The total number of 'no acknowledgement' signaling messages received by the MTA. Set the **arrisMtaDevResetCallStats** object to **true**(1) to clear this counter.

arrisMtaDevSignalingLastMsg1–16

(read-only) These objects contain a 255-byte segment of the CallP last signaling message sent or received. The sixteen objects together can display a signaling message as large as 4000 bytes. If the message does not require all sixteen objects, then the empty objects display the value "Buffer is empty." Use **arrisMtaDevEnableCallSigLastMsgRpt** to enable or disable reporting of the CallP signaling last message.

arrisMtaDevCallStatsTable

This table reports various end of call statistics. All objects in this table are read-only, and are indexed by line number.

arrisMtaDevCallStatsRtpTxPkts

The total number of RTP packets sent from the endpoint during the most recent call. This value is obtained from the signaling end-of-call statistics.

arrisMtaDevCallStatsRtpRxPkts

The total number of RTP packets received by the endpoint during the most recent call. This value is obtained from the signaling end-of-call statistics.

arrisMtaDevCallStatsRtpPercentPktsLost

The percentage of RTP packets lost during the most recent call. This value is obtained from the signaling end-of-call statistics.

The value of this object is expressed in increments of 1/100 percent. For example, a value of **1745** means that 17.45% of the packets were lost.

arrisMtaDevCallStatsAvgJitter

The average jitter measurement, in milliseconds, during the most recent call. This value is obtained from the signaling end-of-call statistics.

arrisMtaDevCallStatsMaxJitter

The maximum jitter measurement, in milliseconds, during the most recent call. This value is obtained from DSP statistics.

arrisMtaDevCallStatsAvgLatency

The average latency, in milliseconds, observed during the most recent call. This value is obtained from the RTCP signaling end-of-call statistics.

arrisMtaDevCallStatsHookStatus

The hook status for each endpoint: **onHook**(0) or **offHook**(1).



Note: The disconnected state is not a valid return value for this MIB object.

arrisMtaDevCallStatsSLICStatus

The over temperature condition of the SLIC chips: **normal** (0) or over **temp**(1).

arrisMtaDevCallStatsEndPntOpStatus

The current operational status for each endpoint: **up**(1) (ready to pass packets), **down**(2), or **testing**(3) (in some test mode). This object obtains its value from **ifAdminOperStatus**.

arrisMtaDevCallStatsLineSubState

The current sub-state for each line: **normal** (0), **di agsPendi ng**(1), **di agsFai l ed**(2), **l cProtecti on**(3), or **dspFai l** (4). This object obtains its value from the lineRec.

arrisMtaDevParameters

These objects provide information about E-UE and line parameters. All objects in this group are read-only.

arrisMtaDevMaxCpeAllowed

Reflects the “MaxCpeAllowed” parameter in the CM config file.

arrisMtaDevNetworkAccess

Reflects the “NetworkAccess” parameter set in the CM config file: **no**(0) or **yes**(1).

arrisMtaDevQosMode

Reflects the setting of the DSx DQoS bit (0x00004000) in the CallP Feature Switch: **bestEffort-FullDQoS-PCMM**(0) or **dsxMode**(1).

arrisMtaDevEventFormat

The PacketCable event format in use: **pktc10**(0) (PacketCable 1.0) or **pktc15**(1) (PacketCable 1.5).

arrisMtaDevLineParameterTable

This table reports various line parameters on the eMTA.

arrisMtaDevInterfaceIndex

The **ifIndex** object for a particular line.

arrisMtaDevPktcDevEvEndpointName

The endpoint name in the following format: **AALN/Line:FQDN/ipaddr**.

arrisMtaDevActiveConnections

The number of active connections for a particular line. Active connections include receive-only, send-only, and send/receive; but not inactive.

arrisMtaDevLineMWIActive

The MWI status for the line.

arrisMtaDevLineRTCPXR

Indicates whether or not RTCP-XR is configured: **disabled**(0) or **enabled**(1). The setting is based on the value of the **arrisMtaDevVqmEnableRemote** and **pktcEDVACodecRTCPXR** objects according to the following table:

arrisMtaDevVqmEnableRemote	pktcEDVACodecRTCPXR(1)	arrisMtaDevLineRTCPXR		
		PC2.0	LegacySIP	NCS
normal	true	enabled	disabled	disabled
normal	false	disabled(2)	disabled(2)	disabled(3)
forceDisable	true	disabled	disabled	disabled
forceDisable	false	disabled	disabled	disabled
forceEnable	true	enabled	enabled	enabled

forceEnable	false	enabled	enabled	enabled
<p>(1)The pkteDVACodecRTCPXR object is applicable only to PC2.0.</p> <p>(2)Though arrisMTADevLineRTCPXR is set to disabled, the remote may still request RTCP-XR to be transmitted which will be honored.</p> <p>(3) Though arrisMTADevLineRTCPXR is set to disabled, the Call Agent may instruct the MTA to request RTCP-XR through NCS signaling.</p>				

arrisMtaDevLineRTCPXRNegotiatedConnectionA

The negotiation status of RTCP-XR for a particular line's A (i.e. first) connection. The status is one of the following:

callNotInProgress(0)

no active call is utilizing this connection/call leg.

notNegotiated(1)

RTCP-XR is not negotiated to be sent on this call leg.

negotiated(2)

RTCP-XR is negotiated to be sent on this call leg.

arrisMtaDevLineRTCPXRNegotiatedConnectionB

The negotiation status of RTCP-XR for a particular line's B (i.e. second) connection. The status is identical to the A connection above.

arrisMtaDevUpSvcFlowParameterTable

This table reports various Upstream Service Flow parameters on the eMTA.

arrisMtaDevDocsQosParamUpSvcFlowSFID

The upstream service flow SFID for a valid service flow index.

arrisMtaDevDocsQosParamUpSvcFlowSchedulingType

The upstream service flow scheduling type for a valid service flow index.

arrisMtaDevVqm Objects

The **arrisMtaDevVqm** MIB provides objects for controlling the Voice Quality Monitoring (VQM) feature and for retrieving VQM data.

The following objects are available.

arrisMtaDevVqmLine

Specifies the line for Voice Quality Metrics reporting.

arrisMtaDevVqmClear

Clears Voice Quality Metrics: **single-line(0)** or **all-lines(1)**. To clear a single line, specify the line using **arrisMtaDevVqmLine**.

arrisMtaDevVqmEnable

Enables or disables recording of Local Voice Metrics: **disable(0)** or **enable(1)** (the default).

arrisMtaDevVqmCallNumberTable

Voice Quality Metric history on a per call number basis. This table contains up to ten entries, consisting of the following object:

arrisMtaDevVqmCallNumberIds

Provides a history of call number identifiers for the specified line.

arrisMtaDevVqmCallNumberIdentifier

Specifies the call number for Voice Quality Metrics reporting. Valid IDs are obtained from **arrisMtaDevVqmCallNumberIds**.

arrisMtaDevVqmMetricTable

Voice Quality Metrics pertaining to a particular call number. Entries are indexed by metric number, and contain the following objects:

arrisMtaDevVqmMetricValues

The call data. Walk the **arrisMtaDevVqmMetricTable** to retrieve the data in the buffer.

arrisMtaDevVqmThresholds

The thresholds associated with this metric. The eDVA generates a Syslog message if VQM data scores fall below the specified thresholds.

arrisMtaDevVqmEnableRemote

Determines the policy to control Remote Voice Metrics (XR):

- **normal** (0) (default) — obey directives from the Call Server.
- **forceDisable** (1) — disables XR, overriding any directive from the Call Server.
- **forceEnable** (2) — enables XR, overriding any directive from the Call Server.
- **default** (3) — resets this object to the default and removes persistence.

The value of this object is persistent if it is set after configuration.

arrisMtaDevVqmThresholdEnable

The maximum number of logs allowed to be sent in a report when a threshold is exceeded. The more logs that are allowed, the more metrics that can be reported. A value of **0** (default) disables threshold reporting.

The eDVA reports metrics in the order specified by the **arrisMtaDevVqmMetricIndex**. To minimize network traffic, logs are sent only when a threshold is first exceeded.

Thereafter until the metric returns to normal, logs are inhibited. To send logs each time the threshold is exceeded, add 100 to the value.

See Setting VQM Thresholds for more details.

arrisMtaDevVqmHistorySize

Sets the size of the VQM history buffer. Valid range: **2** to **50** records. Default: **50**.

arrisMtaDevVqmCallNumberIdentifierLastCall

(read-only) The last call identifier for Voice Quality Metrics reporting.

arrisMtaDevOperationalSetup Objects

arrisMtaDevVPNomJitterBuffer

The Voice Playout nominal jitter buffer size, in terms of packetization rate:

packetizationRate1(1) (default), **packetizationRate2**(2), **packetizationRate3**(3), or **packetizationRate4**(4).

arrisMtaDevVPJitterBufferMode

The Voice Playout jitter buffer mode: **adaptive**(1) (default) or **fixed**(2).

arrisMtaDevRTPTxQueueSize

Sets the RTP transmit queue size. Valid range: **2** to **4**. Default: **2**.

arrisMtaDevEchoCancellerTailLength

The length, in milliseconds, of the echo canceller tail: **eightMs**(1) or **thirtyTwoMs**(2) (default). This object can be set only in the eDVA configuration file.

arrisMtaDevDspHandleNonPhaseReversedTone

Configures handling of the DSP non-phase reversed tone detection:

off(1)

ignores tone detection.

onECANenable(2)

(default) the non-phase reversed CED tone is handled with the echo canceler enabled.

onECANDisabled(3)

the non-phase reversed CED tone is handled with the echo canceler disabled.

arrisMtaDevProvMethodIndicator

The method used to provision the device. This object can be set only in the configuration file. The following provisioning methods are supported:

docsisOnly(0)

DOCSIS-only provisioning.

fullPacketCable(1)

PacketCable 1.5 flows. The [arrisMtaDevPacketcableProvisioningFlow](#) object specifies which flow is in use.

packetCableMinusKDC(2)

same as full PacketCable, except with IPSEC and SNMPv3 disabled.

singleMAC(5)

single configuration file (SNMPv2, single IP address, single MAC address, no SNMP Informs, IPsec disabled).

basic1(6)

Basic1 SNMPv2, without SNMP Enrollment, Status Informs, or Hash.

basic2(7)

ARRIS Basic2 SNMPv2, without SNMP Enrollment Inform or Hash.

arrisMtaCfgRTPDynPortStart

The starting value for a range of ports that is used dynamically when sending out SIP RTP voice packets. This object and **arrisMtaCfgRTPDynPortEnd** define the port range.

Valid range: **1024** to **65535**. Default: **49152**.

arrisMtaCfgRTPDynPortEnd

The ending value for a range of ports that is used dynamically when sending out SIP RTP voice packets. The value of this object must be higher than **arrisMtaCfgRTPDynPortStart**.

Valid range: **1024** to **65535**. Default: **65535**.

arrisMtaDevVPMMaxJitterBuffer

Indicates the Voice Playout maximum jitter buffer: **packetizationRate1**(1), **packetizationRate2**(2), **packetizationRate3**(3) (default), or **packetizationRate4**(4).

arrisMtaDevPacketCableProvisioningFlow

(read-only) Indicates the PacketCable simplified provisioning flow for PacketCable 1.5 compliance:

secure(0)

PacketCable 1.5 Secure Flow

hybrid2(1)

PacketCable 1.5 Hybrid 2 Flow

hybrid1(2)

PacketCable 1.5 Hybrid 1 Flow

basic2(3)

PacketCable 1.5 Basic 2 Flow

basic1(4)

PacketCable 1.5 Basic 1 Flow

none(5)

ARRIS non-PacketCable 1.5 Flow

arrisMtaDevEnableIndexTenEleven

Set to **enable**(1) to use 10 and 11 as the **ifIndex** for lines 1 and 2. The default behavior uses 9 and 10 for the **ifIndex**.

arrisMtaDevDspCpsSetting

Enables or disables ECAN fast constant power signal detection. The default value of **on**(2) makes the echo canceller disengage immediately when a high level constant power signal is detected.

arrisMtaDevVbdOverwriteLineBitmap

A bitmask that defines which lines use the VbdOverwriteJitterBuffer values for fax/modem jitter buffer settings. A value of **0** (the default) affects no lines; **0x01** enables line 1, and so on.

arrisMtaDevVbdOverwriteMinJitterBuffer

When **arrisMtaDevVbdOverwriteLineBitmap** is set for the line, this value is used as the minimum jitter buffer setting in all modem/fax calls. Valid range: **10** to **160**. Default: **20**.

arrisMtaDevVbdOverwriteNomJitterBuffer

When **arrisMtaDevVbdOverwriteLineBitmap** is set for the line, this value is used as the nominal jitter buffer setting in all modem/fax calls. Valid range: **10** to **160**. Default: **20**.

arrisMtaDevVbdOverwriteMaxJitterBuffer

When **arrisMtaDevVbdOverwriteLineBitmap** is set for the line, this value is used as the maximum jitter buffer setting in all modem/fax calls. Valid range: **10** to **160**. Default: **20**.

arrisMtaDevEventHideFQDNandIPAddress

Set to **enable(1)** to hide the MTA FQDN and IP address on the Event Log web page. Logging into the Advanced pages allows an operator to view this information.

arrisMtaDevDhcpOptionOverride

Set to **on(2)** to disable DHCP option code 122/177 sub-option 3 value enforcement.

The default value, **off(1)**, enforces DHCP option 122/177 suboption 3 value comparison checking. Typically, the value received in the DHCP OFFER should not change in the DHCP ACK. DHCP RENEW/REBIND values should be consistent with the value received in the DHCP OFFER.

arrisMtaDevDefaultReasonNoCIDName

The reason sent to the CPE when the Caller ID Name is not included in the signal request. The default value is **unavailable(0)**. The following table shows the reason sent for each setting.

Value	Reason	Description
unavailable(0)	'O'	Out of area
private(1)	'P'	Private caller
sendnothing(2)	NULL	No reason sent
sdmf(3)	number	Number in NA SDMF format
excludeName(4)	nothing	No name parameters or reason

arrisMtaDevSipConfigFileURL

The URL of the SIP configuration file for re-downloading provisioning and configuration parameters to this device.

arrisMtaDevSipDwnldConfig

Set to **on(2)** to enable a re-download of the SIP configuration file parameters specified in the configuration file URL obtained from the **arrisMtaDevSipConfigFileURL** object.

arrisMtaDevSpecialConfigurationOverrideEnable

A bitfield that enables proprietary features of the Arris eDVA. Currently, only 0x80000000 is supported, to enable DHCP Option 60 sub-option 18 override. The default value is **0**.

arrisMtaDevRtcpTosValue

The value used in the IP ToS byte for RTCP packets. Valid range: **0** to **63**.

arrisMtaDevAutomaticOsiDelay

The time, in 100ms increments, to wait after a DLCX is received before determining whether an automatic OSI should be generated to force a line disconnect.

Valid range: **0** to **100**. Default: **50** (5 seconds). Use a value of **0** to send OSI immediately (if there are no other connections on the line).



Note: This object takes effect only if the **ppCfgMtaCallpFeatureSwitch** has bit 0x20000000 set.

arrisMtaDevCustomJitterBufferEnabled

Set to **on(1)** to customize jitter buffer settings. When this setting is off (the default), the jitter buffer size is set using the **arrisMtaDevVPNomJitterBuffer** and **arrisMtaDevVPMajJitterBuffer** objects.

The default jitter buffer range depends on the setting of this object:

- When **off(0)**, the defaults are (in ms):
 minimum = [(packet rate * 1) + 5]
 nominal = minimum
 maximum = [(packet rate * 3) + 5]
- When **on(1)**, the defaults are:
 minimum: 5
 nominal: 10
 maximum: 60

arrisMtaDevCustomMinJitterBuffer

The customized voice playout minimum jitter buffer size to use when the **arrisMtaDevCustomJitterBufferEnabled** object is enabled. Valid range: **5** (default) to **160**, in increments of 5.

arrisMtaDevCustomNomJitterBuffer

The customized voice playout nominal jitter buffer size to use when the **arrisMtaDevCustomJitterBufferEnabled** object is enabled. Valid range: **5** to **160**, in increments of 5. Default: **10**.

arrisMtaDevCustomMaxJitterBuffer

The customized voice playout maximum jitter buffer size to use when the **arrisMtaDevCustomJitterBufferEnabled** object is enabled. Valid range: **5** to **160**, in increments of 5. Default: **60**.

arrisMtaDevEnableDHCPLog

Enables or disables eDVA DHCP logging.

arrisMtaDevEnableMGCPLog

Enable or disables CallP signaling logging.

arrisMtaDevClearDHCPLog

Set to **clear**(1) to clear the eDVA DHCP Logs.

arrisMtaDevClearMGCPLog

Set to **clear**(1) to clear the eDVA MGCP Logs.

arrisMtaDevTDDReportToCMS

Enables or disables reporting of TDD detection events to the CMS. The default is **enable**(1).

arrisMtaDevAutomaticCallResourceRecovery

The time, in seconds, to delay after an on-hook event before detecting whether resources acquired while a call was active need to be recovered on a line in the idle state. Set to **0** to disable this feature.

arrisMtaDevOffHookFskDelay

The time, in milliseconds, to delay before sending the FSK to the CPE. The delay starts upon receiving the ACK (DTMF D) from the CPE in response to the CAS tone for Call Waiting (or Type 2) Caller ID. Valid range: **0** to **500**.

arrisMtaDevT38Timeout

The T.38 timeout, in seconds. The audio is muted for this period before reporting T.38 failure events. Valid range: **1** to **30**. Default: **15**.

arrisMtaDevSuperG3FaxRelay

Set to **enable**(1) to allow SuperG3 fax processing upon detection of the V.21 CM or V.8 data signal. When enabled, the MTA handles the detection of SuperG3 signaling to start the T.38 process. The negotiation during call setup determines whether or not T.38 can be used to send the fax.

When disabled (the default), the MTA still detects signaling for SuperG3 fax, and the signal is used to setup the endpoint for SuperG3 pass-thru fax transmission via G.711.

arrisMtaDevDTMFEndEventForceAscending

Enables or disables RFC 2833 DTMF end event duration force ascending. The default is disabled.

arrisMtaDevDspHandleBellModemTone

Set to **enable**(1) to detect the Bell Modem Tone (2225 kHz). The Bell Modem Tone is frequently used by older data modems, usually in low speed setups. When enabled, the DSP detects the Bell Modem Tone from either the local or network end. When disabled (the default), the tone is ignored.

arrisMtaDevDhcpSubOpt3Immediate

Set to **on**(2) to enable immediate comparison and handling of MTA DHCP Option 122 sub-option 3. SNMP notifications are sent to the new Provisioning SNMP Entity. The default value is **off**(1).

arrisMtaDevMaxCallPServiceFlows

Used to limit the number of active calls (service flows). Outgoing calls, incoming calls, and conference call legs are included in this count. The eDVA ports can call each other without this limitation. Valid range: **0** to **64**. The default value is **0**.

arrisMtaDevCmlpTable

Provides a way to read the CM IP address from the eDVA. The table has one entry, containing the following objects:

arrisMtaDevCmlpAddressType

The CM IP address type: **ip v4**(1) or **ip v6**(2).

arrisMtaDevCmlpAddress

The CM IP address.

arrisMtaDevCmlpPhysAddress

The CM MAC address.

arrisMtaDevEndPntTable Objects

This table provides per-line provisioning details.

arrisMtaDevEndPntDialingMethod

The method used to dial the digits for this endpoint:

tone(1)

(default) tone dialing (DTMF)

pulse(2)

dial-pulse signaling (DTMF is disabled)

toneAndPulse(3)

tone dialing (DTMF) and dial-pulse signaling

pulseWithDTMFRelay(4)

DTMF is disabled, and pulse dialed digits are relayed in-band to the media gateway

toneAndPulseWithDTMFRelay(5)

DTMF is enabled, and pulse dialed digits are relayed in-band to the media gateway.



Note: The values **pulseWithDTMFRelay**(4) and **toneAndPulseWithDTMFRelay**(5) require an IPDT solution as well as DTMF support by the media gateway.

arrisMtaDevEndPntRingingWaveform

The voltage waveform used when ringing this endpoint: **normal** (1) (default) or **sinusoidal** (2). Sinusoidal waveform is for use with telephones that exhibit increased sensitivity to ring voltage waveform.

arrisMtaDevEndPntFaxOnlyLineTimeout

Set to a non-zero value to define a line as a fax-only line for NCS. The default value is **0**.

In fax-only mode, this object defines a timer (in seconds) that is started after the RTP mode becomes sendReceive. If this timer expires without detecting fax/modem tones on the connection, the call is dropped.

arrisMtaDevPersistentLineStatus

Controls persistent line status. This object and **ifAdminStatus** impact the line status.

- Set to **ignore**(0) (the default) to base the line status on the setting of **ifAdminStatus** in the MTA configuration file. If **ifAdminStatus** is not set in the MTA configuration file, then it uses the default value: **up**(1) if the line is provisioned or **down**(2) if the line is not provisioned.
- Set to **forceDisable**(1) to force the service status of the line to **down**(2) and the line state to EP_OOS after a reset.

This object is always ignored in a configuration file.

arrisMtaDevEndPntCallWaitingRepeatSteady

Set to **enabled**(1) to repeat the call waiting tone forever. The default value **disabled**(0) uses the normal repeat rules.

arrisMtaDevEndPntCIDEnable

Set to **disabled**(0) to disable sending of all Caller ID fields.

arrisMtaDevEndPntCIDNameEnable

Set to **disabled**(0) to disable sending the CallerID Name field.

arrisMtaDevEndPntCIDDateTimeEnable

Set to **disabled**(0) to disable sending the CallerID Date/Time field.

arrisMtaDevEndPntLoopReversal

When set to **enabled**(1), the line reverses to normal polarity once the originating party hangs up first. The default value, **disabled**(0), maintains reverse loop polarity while the originating party hangs up.

arrisMtaDevEndPntGainControlTxVoice

The transmit digital gain adjustment, in dBm, for voice calls. When set to a value other than **disabled**(-128), this object supersedes **arrisMtaDevGainControlTxVoice**. Valid range: **-16** to **16**. Default: **0**. A value of **2** increases the voice level by 2 dBm. A value of **-2** decreases the voice level by 2 dBm.

This object does not affect the levels of local tones or FSK.



Note: Use caution when changing this object. Increasing or decreasing the voice level by the larger numbers allowed in the range may compromise voice quality.

arrisMtaDevEndPntGainControlRxVoice

The receive digital gain adjustment, in dBm, for voice calls. When set to a value other than **disabled**(-128), this object supersedes **arrisMtaDevGainControlRxVoice**. Valid range: **-16** to **16**. Default: **0**. A value of **2** increases the voice level by 2 dBm. A value of **-2** decreases the voice level by 2 dBm.

This object does not affect the levels of local tones or FSK.



Note: Use caution when changing this object. Increasing or decreasing the voice level by the larger numbers allowed in the range may compromise voice quality.

arrisMtaDevGainControl Objects

These objects provide digital gain adjustments for each endpoint.

arrisMtaDevGainControlFSK

The transmit digital gain adjustment, in dBm, for MTA-generated FSK tones (CID and VMWI). Valid range: **–10** to **2**. Default: **0**.

arrisMtaDevGainControlCAS

The transmit digital gain adjustment, in dBm, for MTA-generated CAS tones. Valid range: **–2** to **2**. Default: **0**.

arrisMtaDevGainControlLocalTone

The transmit digital gain adjustment, in dBm, for all MTA-generated Call Progress tones (Dialtone, Busytone, Ringback, etc.) toward the CPE. Valid range: **–2** to **2**. Default: **0**. This object does not effect CAS tone levels; use [arrisMtaDevGainControlCAS](#) to adjust the CAS tone.

arrisMtaDevGainControlNetworkTone

The transmit digital gain adjustment, in dBm, for MTA-generated Call Progress tones toward the network (Ringback). Valid range: **–2** to **2**. Default: **0**.

arrisMtaDevGainControlLocalDTMF

The transmit digital gain adjustment, in dBm, for MTA-generated DTMF tones toward the local CPE (ex. DTMF CID). Valid range: **–15** to **9**. Default: **0**.

arrisMtaDevGainControlNetworkDTMF

The transmit digital gain adjustment, in dBm, for MTA-generated DTMF tones toward the network (ex. IPDT Pulse Dialing). Valid range: **–9** to **9**. Default: **0**.

arrisMtaDevGainControlTxVoice

The transmit digital gain adjustment, in dBm, for voice. This value does not effect the levels of local tones or FSK. Valid range: **–16** to **16**. Default: **0**.



Note: Use caution when changing this object. Increasing or decreasing the voice level by the larger numbers allowed in the range may compromise voice quality.

arrisMtaDevGainControlRxVoice

The receive digital gain adjustment, in dBm, for voice. Valid range: **–16** to **16**. Default: **0**.



Note: Use caution when changing this object. Increasing or decreasing the voice level by the larger numbers allowed in the range may compromise voice quality.

arrisMtaDevLevelControl Objects

These objects control off-hook tone gain.

arrisMtaDevLevelControlOffHookEnable

Set to **enable(1)** to allow use of the [arrisMtaDevLevelControlOffHookFSK](#) and

arrisMtaDevLevelControlOffHookCAS objects, instead of **arrisMtaDevLevelControlFSK** and **arrisMtaDevLevelControlCAS**, in off-hook situations.

arrisMtaDevLevelControlOffHookFSK

The transmit digital gain setting, in dBm, for MTA-generated FSK tones (CID and VMWI) while the line is off-hook. Valid range: **-32** to **-10**. Default: **-15**.

arrisMtaDevLevelControlOffHookCAS

The transmit digital gain setting, in dBm, for MTA-generated CAS tones (CID and VMWI) while the line is off-hook. Valid range: **-32** to **-10**. Default: **-15**.

arrisMtaDevDiagLoopTable Objects

These objects provide per-line Loop Diagnostic (patent pending) details.

arrisMtaDevDiagLoopTime

(read-only) The time and date when loop diagnostics were last run on the selected line.

arrisMtaDevDiagLoopRequest

Set this value to **true(2)** to start loop diagnostics on the line.

arrisMtaDevDiagLoopLastResult

(read-only) Current loop diagnostics status; one of:

- **diagnosti cs- passed(1)**
- **hazardous- potenti al - test- fai lure(2)**
- **forei gn- emf- test- fai lure(3)**
- **resi sti ve- faul ts- test- fai lure(4)**
- **recei ver- offhook- test- fai lure(5)**
- **ri nger- test- fai lure(6)**
- **i nval i d- state- to- i ni t- di ags(7)**
- **l i ne- i s- unprovi si oned(8)**
- **diagnosti cs- resul ts- pendi ng(9)**
- **not- started(10)**
- **unsupported(11)**
- **ri nger- test- warni ng(12)**

When complete, the status is either **diagnosti cs- passed(1)** or the first test failed. During the test, the status is always **diagnosti cs- resul ts- pendi ng(9)**.

arrisMtaDevDiagLoopHazardousPotentialTest

(read-only) The Hazardous Potential Test result for the last time that loop diagnostics were run.

arrisMtaDevDiagLoopForeignEmfTest

(read-only) The Foreign EMF Test result for the last time that loop diagnostics were run.

arrisMtaDevDiagLoopResistiveFaultsTest

(read-only) The Resistive Faults Test result for the last time that loop diagnostics were run.

arrisMtaDevDiagLoopReceiverOffHookTest

(read-only) The Receiver Off-hook Test result for the last time that loop diagnostics were run.

arrisMtaDevDiagLoopRingerTest

(read-only) The Ringer Test result for the last time that loop diagnostics were run.

ARRIS-MTA-MIB (battery telemetry items)

Battery-related objects appear in the following groups:

- **arrisMtaDevPwrSupplyBase**
- **arrisMtaDevPwrSupplyControl**
- **arrisMtaDevPwrSupplyTimers**
- **arrisMtaDevPwrSupplyStats**
- **arrisMtaDevPwrSupplyAlarm**

arrisMtaDevPwrSupplyBase Objects

These read-only objects provide basic information.

arrisMtaDevBatteryChargerFWRev

The firmware revision of the battery charger.

arrisMtaDevPwrSupplyControl Objects

These objects provide control and status information for the charger system.

arrisMtaDevPwrSupplyEnableDataShutdown

Set to **disabled**(2) to allow CPE interfaces (Ethernet and wifi, if equipped) to provide service during a power outage. This reduces battery hold-up time.

arrisMtaDevPwrSupplyLowBatteryThresh

Sets the low battery threshold, in 10 watt*minutes increments. A charge below this threshold indicates a low battery condition. The initial default value is equivalent to 1 hour of holdup time.

arrisMtaDevPwrSupplyTypicalIdlePwr

(read-only) The typical idle power, in 50 mW increments. A nominal value is loaded when the modem powers up. This value is used in conjunction with the Tested Battery Capacity, Low Battery Threshold, and the Replace Battery Threshold to determine when to raise the Replace Battery or Low Battery alarms.

arrisMtaDevPwrSupplyReplaceBatThresh

The minimum acceptable battery charge, in 10 watt*minutes increments, needed to achieve the desired End of Life hold-up time, based on typical idle power. If the Tested Battery Capacity minus the Charge Hysteresis loss (20%) is less than this value, the modem raises the Replace Battery alarm. The initial default value provides 1 hour of holdup time.

arrisMtaDevPwrSupplyChargeState

The “full charge” state, in 10 watt*minutes increments. The initial default value is 80% of the Rated Battery Capacity. Writing to this object forces the charger to start a discharge/charge cycle, to charge the battery to the specified level.

Be careful when changing this value. Specifying a higher charge level may increase hold-up times, but can reduce overall battery life.

arrisMtaDevPwrSupplyBatteryTest

Controls the battery test schedule:

testScheduled(0)

Resumes the battery test scheduler at its current value. When read, this value indicates that the battery test runs when the scheduled time expires.

disableAutoTesting(1)

Freezes the battery test scheduler at its current value. When read, this value indicates that the battery test is suspended. Removing and replacing the battery forces the charger to run the test one time. To resume normal operation, set the value to **testScheduled(0)** or **testInProgress(2)**.

testInProgress(2)

Starts the battery test cycle immediately, and resets the test scheduler to its default value of 180 days. When read, this value indicates a test in progress.

testPending(3)

(read-only) A battery test was in progress, and either AC power was lost or a full charge was requested.

arrisMtaDevPwrSupplyConfigRunTime

The estimated battery hold-up time, in minutes, based on the typical idle power and the programmed battery charge setting. The battery hold-up time may be adjusted using this object. By setting the holdup time to a lower value, the total service life of the battery is extended. Increasing the hold-up time decreases the total service life of the battery.



Note: This value can only be set in multiples of 5 minutes. Setting a value greater than **arrisMtaDevPwrSupplyBatAvailableMinutes** does not extend the hold-up time beyond that specified by **arrisMtaDevPwrSupplyBatAvailableMinutes**.

arrisMtaDevPwrSupplyConfigReplaceBatTime

The replace battery threshold, in terms of minutes of hold-up time. If a battery’s capacity has degraded to a point where its hold-up time is below this threshold, the Replace Battery condition becomes active.

This value can only be set to multiples of 5 minutes. The default value at power up is 60 minutes.

arrisMtaDevPwrSupplyOverTempAlarmControl

Set to **enable(1)** to issue an Over Temperature Alarm if the charger exceeds the temperature specified in **arrisMtaDevPwrSupplyOverTempAlarmThreshold**, and to shut down the charger if the temperature exceeds 90°C.

arrisMtaDevPwrSupplyOverTempAlarmThreshold

The temperature threshold, in degrees C, for the Over Temperature alarm. Valid range: **50** to **70**.

arrisMtaDevPwrSupplyTemperature

(read-only) The current charger temperature, in degrees C. This is available only when **arrisMtaDevPwrSupplyOverTempAlarmControl** is enabled.

arrisMtaDevPwrSupplyHiTempBatteryShutdownControl

Set to **enable(1)** to turn off the battery if the temperature reaches 75°C.

arrisMtaDevPwrSupplyHighestTemperature

(read-only) The highest recorded battery charger temperature, in degrees C. This is available only when **arrisMtaDevPwrSupplyOverTempAlarmControl** is enabled.

arrisMtaDevPwrSupplyHighestTemperatureTime

(read-only) The time and date when the highest temperature was recorded.

arrisMtaDevPwrSupplyHighestTemperatureClear

Set to **clear(1)** to clear current values of **arrisMtaDevPwrSupplyHighestTemperature** and **arrisMtaDevPwrSupplyHighestTemperatureTime**. These objects reset to current values within 4 seconds.

arrisMtaDevPwrSupplyControlChargerReset

Set to **true(1)** to reset the battery charger.



Note: Resetting the battery charger during an AC Fail condition immediately shuts down the unit.

arrisMtaDevPwrSupplyTimers Objects

These objects provide timers for the battery charging system.

arrisMtaDevPwrSupplyDataShutdownTime

The timeout period, in seconds, until the device terminates data services (Ethernet, USB, and wifi if equipped) after loss of AC power. The default value depends on the gateway type:

- 802.11ac-capable Gateway products: 30 seconds.
- all others: **900** seconds (15 minutes).

arrisMtaDevPwrSupplyFullChargeTime

The number of days to maintain the battery at 100% of its rated voltage. Any ongoing or schedule battery tests are stopped. This may be useful when widespread power outages are expected (for example, an approaching storm system). After the time has elapsed, the charger allows the battery to return to the value specified by **arrisMtaDevPwrSupplyChargeState**.

Valid range: **1** to **16**. A value of **0** may also be read, which indicates that the charger is operating normally.

arrisMtaDevPwrSupplyStats Objects

These object provide battery and charger statistics. All objects in this group are read-only.

arrisMtaDevPwrSupplyBatteryTestTime

The present value of the test timer scheduler, in days. If the value is 0xFF (255), the timer has been paused.

arrisMtaDevPwrSupplyRatedBatCapacity

The rated capacity, in 10 watt*minutes increments, of the battery.

arrisMtaDevPwrSupplyTestedBatCapacity

The measured battery capacity, in 10 watt*minutes increments, as measured by the last battery test cycle.

arrisMtaDevPwrSupplyBatStateOfCharge

The present battery state of charge, in 10 watt*minutes increments. This value is approximate and is re-calibrated following a battery test cycle.

arrisMtaDevPwrSupplyReadBatteryPwr

The present load power, in 50mW increments, over an eight-second moving average. This is the power being removed (when running on battery power) or applied to the battery (when charging).

arrisMtaDevPwrSupplySecondsOnBattery

The time, in seconds, that the modem has been using battery power.

arrisMtaDevPwrSupplyBatRatedMinutes

The estimated rated hold-up time, in minutes, based on typical idle power and the rated capacity of the battery when fully charged.

arrisMtaDevPwrSupplyBatAvailableMinutes

The estimated available hold-up time, in minutes, based on typical idle power and the tested capacity of the battery when fully charged.

arrisMtaDevPwrSupplyTelemetryValues

Power supply telemetry values, used by ARRIS technical support when troubleshooting a battery issue.

arrisMtaDevBatteryStatusTable

The Power Supply telemetry table. All objects in this table are read-only.

arrisMtaDevBatteryOperState

The current operational state of the battery:

- **unavailable(0)**
- **invalid(1)**
- **shutdownWarning(2)**
- **batteryReversedShorted(3)**
- **batteryLow-replaceBattery-acFail(4)**
- **batteryLow-replaceBattery(5)**

- **batteryLow-acFail** (6)
- **batteryLow**(7)
- **batteryMissing**(8)
- **acFail-replaceBattery**(9)
- **replaceBattery**(10)
- **acFail** (11)
- **normal** (12)
- **testInProgress**(13)
- **chargerFailure**(14)

arrisMtaDevBatteryLastStateChange

The value of **sysUpTime** when the battery entered its current operational state.

arrisMtaDevBatteryOperSubState

The current sub-state of the battery. The sub-state is not designed to match the **arrisMtaDevBatteryOperState** but to provide additional information about the charger status.

arrisMtaDevBatteryOrderingCode

The ARRIS ordering code for the battery.

arrisMtaDevBatteryEprom

EPROM information for the battery.

Administration

Administration involves collecting performance statistics, capacity planning, and maintaining system reliability.

Administration Objects

Touchstone supports both standard and ARRIS-proprietary SNMP MIBs for administration and other purposes. This section describes several generic objects. See the Operations chapter for ARRIS-proprietary objects.

System Description Objects

The **system** objects can be used to identify a Touchstone product and find general information about it. Most of these objects are generic to all devices, but those described below have values unique to ARRIS products.

sysDescr Object

Touchstone firmware supports the **sysDescr** MIB object. This object provides firmware version and product description information in the format specified in section 4.2.1 of the *DOCSIS Operations Support System Interface Specification*, CM-SP-OSSlv3.0-I15-100115. The specification requires the **sysDescr** object to be in the following format:

```
any text <<HW_REV: hardware version information;
VENDOR: vendor name; BOOTR: BootROM version;
SW_REV: firmware version; MODEL: hardware model information>> any text
```

Since the content is in a consistent format, the object can be automatically parsed and used for various functions such as determining when firmware upgrades are required. The fields in the **sysDescr.0** object are:

Field	Meaning	Description
HW_REV	Hardware Revision	The hardware revision of the Telephony Modem. ARRIS updates this field as needed to reflect significant hardware changes or improvements to the product.
VENDOR	Vendor Name	The vendor name; in this case, "Arris Interactive, L.L.C."
BOOTR	Boot ROM	The BootROM image version that is embedded in the product, and used to load the application firmware image.

Field	Meaning	Description
SW_REV	Firmware Revision	The firmware version of the application firmware image currently loaded on the device. Note: In addition to the System Descriptor MIB object (sysDescr.0), the SW_REV information is also available in the docsDevSwCurrentVers MIB object.
MODEL	Model Number	The hardware model number of the device.

The following is an example of the [sysDescr.0](#) contents for a modem device.

```
ARRIS DOCSIS 3.1 / PacketCable 1.5 Touchstone Residential Gateway <<HW_REV:
4; VENDOR: ARRIS Group, Inc.; BOOTR: 1.2.8.491938; SW_REV: 01.01.086.01;
MODEL: TG3452A>>
```

sysORTable Objects

The following [sysORTable](#) objects provide useful information:

[sysORID](#)

Equivalent to the OID representing [modemAgentDocsis20](#).

[sysORDescr](#)

Contains the string “DOCSIS 3.0 Cable Modem agent.”

sysObjectID Object

The [sysObjectID](#) object provides a condensed version of the data found in the [sysDescr](#) object. The data in this object is in OID format, consisting of the following fields:

Field(s)	Description
1–9	Equivalent to the OID representing arris : 1.3.6.1.4.1.4115 . Some SNMP software may display this as arris or enterprises.4115 .
10	The model number; for example, 802
11	Hardware release
12–16	Not used, display as 0

Bridging and Routing Objects

Depending on provisioning and model-specific capabilities, Touchstone hardware can function either as a bridge or a router. The MIBs described here provide information about each function.

dot1dBridge Objects

The **dot1dBridge** MIB provides information about the bridging function. The following objects can be of use:

dot1dBaseBridgeAddress.0

In Touchstone hardware, the MAC address of the Ethernet interface.

dot1dBaseNumPorts.0

The number of interfaces connected to the bridge. Touchstone hardware uses the **ifIndex** of the interface as indexes to various tables in the **dot1dBridge** MIB.

dot1dBaseType.0

The bridge type.

dot1dTpFdbTable

The forwarding database. This table uses the MAC address of supported interfaces as the index.

rip2 Objects

The **rip2** MIB provides routing information when the E-UE is provisioned for RIPv2 routing support. The following objects can be of use:

rip2IfStatTable

Statistics for each interface enabled for RIP routing.

rip2IfConfTable

RIP interface configuration for each interface enabled for RIP routing.

End of Call Connection Statistics

Touchstone firmware supports the PacketCable 1.0-defined call connection statistics, with clarifications as defined in ECN EC-MGCP-N-04.0175-7. The EC clarifies the requirement to ensure that the statistics represent the actual packets sent/received regardless of the current connection state of the call.



Note: End-of-call connection statistics and Voice Quality Monitoring (VQM) statistics are different features. For information about the VQM feature, see “Managing Voice Quality Monitoring.”

Touchstone firmware makes end-of-call connection statistics available through proprietary MIB objects and Syslog messages. The **arrisMtaDevEnableCallStatsSyslogRpt** MIB object controls end-of-call statistics reporting through Syslog. The **arrisMtaDevEnableCallSigLastMsgRpt** MIB object controls reporting of the last 4K of signaling messages through Syslog. Each object may be set independently, and interact as follows:

CallStats MIB	LastMsg MIB	CMS LoC Alarm	Messages Sent
Disabled	Disabled	Inactive	None
		Active	None
	Enabled	Inactive	None
		Active	None
Enabled	Disabled	Inactive	None
		Active	Last messages sent/received
	Enabled	Inactive	Last 4K of messages sent/received
		Active	Last 4K of messages sent/received

NCS Behavior

Touchstone NCS loads make end-of-call connection statistics available through proprietary MIB objects and Syslog messages. Touchstone firmware supports the PacketCable-defined call connection statistics, with clarifications as defined in ECN EC-MGCP-N-04.0175-7. Previously, the PacketCable NCS specification implied that these statistics were related to the connection mode requested by the CMS. The EC clarifies the requirement to ensure that the statistics represent the actual packets sent/received regardless of the current connection state of the call.

The E-UE sends connection statistics to the call server (and optionally, Syslog servers) during the call tear-down procedure. PacketCable-compliant call servers provide a method of reporting these captured statistics. See the call server documentation for instructions on accessing the statistics.

SIP Behavior

Touchstone SIP loads support end-of-call statistics reporting through the SIP PUBLISH mechanism (defined in RFC 3903). The content of the PUBLISH message uses the session report format.

To enable end-of-call statistics reporting, set the following MIB objects:

pkcEDVACodecPubRepAddrType

Specifies the IP address type (IPv4 or IPv6) of the device that receives statistics reports.

pkcEDVACodecPubRepAddr

The IP address of the device that receives statistics reports. If this object is not specified, end-of-call statistics are disabled.

pkcEDVACodecRTCPXR

Determines whether the eDVA includes far-end statistics in the report. The default is **true(1)**.

End-of-Call Statistics MIB Objects

Touchstone firmware provides MIB objects for monitoring eDVA end-of-call statistics.

The monitoring MIB is broken into groups under **arrisMtaDevTrace**:

- Objects that report on a per-call level (the counter values represent the total count for the most recently completed call on an endpoint). These objects are indexed by the endpoint number in the table **arrisMtaDevCallStatsEntry**.
- Objects that report on a device level (the counter values represent the sum total for all endpoints), located under **arrisMtaDevTrace**.
- Objects that control and reset counters, located under the **arrisMtaDevControl** MIB tree.

Device Level End-of-Call Statistics

Device-level MIB objects represent combined end-of-call statistics for all lines on the eDVA.

arrisMtaDevRtpTxPktsTotal

(read-only) The total number of RTP packets sent from the eDVA since it was last started up or reset. This value represents the total number of packets sent for all endpoints combined. It is calculated from values obtained from the signaling end of call statistics. Set the **arrisMtaDevResetCallStats** object to **true(1)** to clear this counter.

arrisMtaDevRtpRxPktsTotal

(read-only) The total number of RTP packets received by the eDVA since it was last started up or reset. This value represents the total number of packets received for all endpoints combined. It is calculated from values obtained from the signaling end of call statistics. Set the **arrisMtaDevResetCallStats** object to **true(1)** to clear this counter.

arrisMtaDevRtpPercentPktsLostTotal

(read-only) The percentage of RTP packets lost since the eDVA was last started up or reset. This value represents the total number of packets lost for all endpoints combined. It is calculated from values obtained from the signaling end of call statistics. Set the **arrisMtaDevResetCallStats** object to **true(1)** to clear this counter.

The value of this object is expressed in increments of 1/100 percent. For example, a value of **1745** means that 17.45% of the packets were lost.

arrisMtaDevRtpPktsLostTotal

(read-only) The number of RTP packets lost since the MTA was last started up or reset. This value represents the total number of packets lost for all endpoints combined. It is calculated from values obtained from the signaling end of call statistics. Set the **arrisMtaDevResetCallStats** object to **true(1)** to clear this counter.

arrisMtaDevLastCallStartTime

(read-only) The last call start time from the eDVA.

arrisMtaDevLastCallEndTime

(read-only) The last call end time from the eDVA.

arrisMtaDevSignalingAvgLatency

(read-only) The average latency or delay, in milliseconds, for responses to signaling messages. It is calculated from values obtained from the signaling end of call statistics. Set the **arrisMtaDevResetCallStats** object to **true(1)** to clear this counter.

arrisMtaDevSignalingTxSuccessfulMsgCnt

(read-only) The total number of successful signaling messages sent from the eDVA. Set the **arrisMtaDevResetCallStats** object to **true(1)** to clear this counter.

arrisMtaDevSignalingRxSuccessfulMsgCnt

(read-only) The total number of successful signaling messages received by the MTA. Set the **arrisMtaDevResetCallStats** object to **true(1)** to clear this counter.

arrisMtaDevSignalingTxNAKCnt

(read-only) The total number of negative acknowledgement signaling messages (NAKmessages) sent from the MTA. Set the **arrisMtaDevResetCallStats** object to **true(1)** to clear this counter.

arrisMtaDevSignalingRxNAKCnt

(read-only) The total number of negative acknowledgement signaling messages (NAKmessages) received by the MTA. Set the **arrisMtaDevResetCallStats** object to **true(1)** to clear this counter.

Line Level Statistics

Line-level MIB objects, collected in the **arrisMtaDevCallStatsTable**, report end-of-call statistics and general status for the most recently-completed call on each line in the eDVA. Entries in this table are indexed by the endpoint number (1 to N), where N is the maximum number of lines supported by the eDVA; for example, **arrisMtaDevCallStatsAvgJitter.2** shows the average jitter for the second line.

Other MIB objects in this table provide information about endpoint temperature, hook status, and operational status.

arrisMtaDevCallStatsRtpTxPkts

The total number of RTP packets sent from the endpoint during the most recent call. This value is obtained from the signaling end-of-call statistics.

arrisMtaDevCallStatsRtpRxPkts

The total number of RTP packets received by the endpoint during the most recent call. This value is obtained from the signaling end-of-call statistics.

arrisMtaDevCallStatsRtpPercentPktsLost

The percentage of RTP packets lost during the most recent call. This value is obtained from the signaling end-of-call statistics.

The value of this object is expressed in increments of 1/100 percent. For example, a value of **1745** means that 17.45% of the packets were lost.

arrisMtaDevCallStatsAvgJitter

The average jitter measurement, in milliseconds, during the most recent call. This value is obtained from the signaling end-of-call statistics.

arrisMtaDevCallStatsMaxJitter

The maximum jitter measurement, in milliseconds, during the most recent call. This value is obtained from DSP statistics.

arrisMtaDevCallStatsAvgLatency

The average latency, in milliseconds, observed during the most recent call. This value is obtained from the RTCP signaling end-of-call statistics.

arrisMtaDevCallStatsHookStatus

The hook status for each endpoint: **onHook**(0) or **offHook**(1).



Note: The disconnected state is not a valid return value for this MIB object.

arrisMtaDevCallStatsSLICStatus

The over temperature condition of the SLIC chips: **normal** (0) or over **temp**(1).

arrisMtaDevCallStatsEndPntOpStatus

The current operational status for each endpoint: **up**(1) (ready to pass packets), **down**(2), or **testing**(3) (in some test mode). This object obtains its value from **ifAdminOperStatus**.

arrisMtaDevCallStatsLineSubState

The current sub-state for each line: **normal** (0), **diagsPendi ng**(1), **diagsFai l ed**(2), **lcProtecti on**(3), or **dspFai l** (4). This object obtains its value from the lineRec.

Clearing Counters

To clear the device-level end-of-call statistics counters, set the **arrisMtaDevResetCallStats** MIB to **true**(1). Setting this object to a value of **false**(2) has no effect. Reading this object always returns **false**(2).

This object clears the following counters:

- **arrisMtaDevRtpTxPktsTotal**
- **arrisMtaDevRtpRxPktsTotal**
- **arrisMtaDevRtpPercentPktsLostTotal**
- **arrisMtaDevSignalingAvgLatency**
- **arrisMtaDevSignalingTxSuccessfulMsgCnt**
- **arrisMtaDevSignalingRxSuccessfulMsgCnt**
- **arrisMtaDevSignalingTxNAKCnt**
- **arrisMtaDevSignalingRxNAKCnt**
- **arrisMtaDevSignalingRxNoACKCnt**

Last Signaling Message Sent

To make a short trace of the last messages available at the end of the most recent call, Touchstone firmware allocates a 4K byte circular buffer to store the last signaling messages sent on the most recent call. To enable collection of signaling messages, set the [arrisMtaDevEnableCallSigLastMsgRpt](#) object to **enable(1)**. The default value is **disable(0)**.

The following table shows the behavior of the buffer under certain conditions.

arrisMtaDevEnableCallSigLastMsgRpt	CMS LoC Alarm	Buffer Contents Stored
Disable	Inactive	None
	Active	Last messages sent/received
Enable	Inactive	Last 4k of messages
	Active	Last 4k of messages

Since the maximum PDU size for a MIB object response is 256 bytes, the message is split into 256-byte segments and stored in MIB objects [arrisMtaDevSignalingLastMsg1](#) through [arrisMtaDevSignalingLastMsg16](#). Empty objects in this group report “Buffer is empty.” When reading these objects, the eDVA sets the [arrisMtaDevEnableCallSigLastMsgRpt](#) object to **disable(0)** to prevent another message from overwriting these MIB objects while reading the contents. After reading the objects, you must manually set the [arrisMtaDevEnableCallSigLastMsgRpt](#) object to **enable(1)** to capture the next message.

Per-Call Syslog Reporting

Touchstone firmware can generate a Syslog report of call status and monitoring information on a per-call basis.

In addition to statistical data, the Syslog report can also include up to the last 4K bytes of signaling messages associated with the last call in a circular buffer. The Syslog report can be configured to supply only statistical data, or statistical data and signaling messages.

Enable or disable Syslog reporting using the [arrisMtaDevEnableCallStatsSyslogRpt](#) object, which is part of the [arrisMtaDevBase](#) MIB. The default value for this object is **disable(0)**; set to **enable(1)** to enable reporting.

Enable or disable the signaling message buffer using the [arrisMtaDevEnableCallSigLastMsgRpt](#) object. The default value for this object is **disable(0)**; set to **enable(1)** to enable the buffer. The contents of the buffer depends on the setting of this MIB, the [arrisMtaDevEnableCallStatsSyslogRpt](#) object, and the state of the CMS LoC alarm.

The following table shows how the MIB settings interact with the Loss of Comms alarm to affect the buffer contents.

arrisMtaDevEnable CallStatsSyslogRpt	arrisMtaDevEnable CallSigLastMsgRpt	CMS LoC Alarm	Buffer contents contained in Syslog
Disable	Disable	Inactive	None
		Active	None
	Enable	Inactive	None
		Active	None
Enable	Disable	Inactive	None
		Active	Last messages sent/received
	Enable	Inactive	Last 4k of messages
		Active	Last 4k of messages

Per Call Syslog Report Example

Below is an example of a per-call Syslog report (statistical output only):

```
09-06-2007      10:11:02      Local 0. Warning      10.1.36.205
Sep 06 09:39:27 2007 AALN/1: mta205.dev36/10.1.36.205 <6> <4115> <37>
<00:00:CA:CB:23:E7> <Call Stats: L1, HW: TM402G, SW: 5.2.21, RTP Tx 0,
RTP Rx 0, RTP Lost 0.00%, Prov State: MTA Prov Complete. (20),
Avg Jtr 0, Max Jtr 0, Avg Ltc RTP 0, Avg Ltc Sig Msg 11, No ACKs 0,
Batt 100%: INACTIVE, CMS LOC: INACTIVE>
```

The fields in the Syslog report are as follows:

- HW: Hardware Model Number
- SW: Firmware Rev Version Number
- Ttl Rx RTP: Total number of RTP packets received. This only reports RTP packets and not data packets.
- Ttl Tx RTP: Total number of RTP packets sent. This only reports RTP packets and not data packets.
- RTP Lost: Percentage of RTP packets lost. This only reports RTP packets and not data packets.
- Prov State: The step of the eDVA Provisioning Flow that the eDVA is currently in.
- Avg Jtr: Average Jitter in milliseconds.
- Max Jtr: Maximum Jitter in milliseconds.
- Avg Ltc RTP: Average latency for RTP packets, in milliseconds.
- Avg Ltc Sig Msg: average latency for a response to signaling messages, in milliseconds. This value is calculated from end-of-call signaling statistics and is a running average during eDVA uptime.
- No ACKs: Count of the number of “negative acknowledgment” messages received from the Call Agent.
- SNMP Traps: Battery Percentage; CMS LOC.

Message Trace Example

The following is an example Syslog report showing Send (Xmit) and Receive (Rcv) Messages:

```
Nov 17 18:34:11 2005 mta161.dev35 <13>  
<4115> <42> <00:00:CA:CB:22:FB> <Rcv: (3: 1 of 1) -  
<010>Q: loop<010>R: hf(I), hu(N)'>  
Nov 17 18:34:11 2005 mta161.dev35 <14> <4115> <41> <00:00:CA:CB:22:FB>  
<Xmit: (3: 1 of 1) - '200 31855 OK'>
```

See see "[Capturing Signaling Traces](#) (page 188) for a breakdown of send and receive message formats.

Using the Speedtest Application

The Speedtest application allows remote testing of download and upload speeds between the modem and an external server, without using a CPE.

Measuring downstream and upstream speed is done by transferring a file between a network server and the HTTP/FTP client embedded in the modem. Average throughput is calculated from the size of the file transferred and the total transfer time.

Speed tests use the TR-069 management interface, supporting the TR-143 Diagnostics parameters. AR01.1 stores the results of the last test until another test runs, or the Gateway is rebooted.

Server Requirements

To conduct speed tests, an HTTP or FTP server must be available. Server location can be used to determine throughput in a variety of ways:

Server Location	Throughput Test
Local headend	HFC throughput
Master headend	Internal network throughput
External site	Peering throughput

An HTTP server used for Uplink testing must meet the following requirements:

- HTTP 1.1 support
- Support for an upload submission form, with an input element of type FILE
- The server must provide a script to process POST HTTP requests, in accordance with RFC 1867

The open-source Apache server supports all requirements and is widely available.

Running a Speed Test using TR-143 Objects

1. Configure one or both of the speed tests as follows:

Download test:

- Device.DownloadDiagnostics.DownloadURL: an HTTP or FTP URL, pointing to a test file. A large file, such as a video, is recommended.

Upload test:

- Device.UploadDiagnostics.UploadURL: an HTTP or FTP URL, specifying the destination for the test file.
- Device.UploadDiagnostics.TestFileLength: the amount of data, in bytes, to transfer. A large value, 1MB or more, is recommended.

2. To start the test:

- Download test: set Device.DownloadDiagnostics.DiagnosticsState to Requested.

- Upload test: set Device.UploadDiagnostics.DiagnosticsState to Requested.
3. Wait for the Gateway to send a "Diagnostics Complete" inform to the ACS.
 4. Read the DiagnosticsState parameter corresponding to the proper test, and verify that the value is "Complete." If the value is "Requested," the test is still running.
 5. Read the parameters for the test results.

If the DiagnosticsState value indicates an error, check your configuration (especially the URLs) to make sure:

- The test is using the correct server.
- The path and file names are correct.
- Any required credentials are correct. For example, FTP URLs require the user name and password to be specified in the form `ftp: //user: pass@i paddr/path/fi l e`.

If problems persist, check for network connectivity issues.

Network Performance Monitoring

Network Performance Monitoring uses the CM component of a Touchstone product to test the performance of individual network legs or end-to-end connectivity. All testing can be performed from the headend or NOC without subscriber intervention.

The tests provide results in JSON format, for easy database entry, parsing, and consolidation.



Note 1: In this release, Network Performance Monitoring supports only IPv4 networks.

Test Types

The following tests are supported:

- webpage download
- DNS latency
- network latency

Setup

Setup is SNMP-based. Add MIB objects from the [arrisNetPerfMonitorMib](#) as needed to configure Network Performance Monitoring.

The following sections describe how to set up Network Performance Monitoring.

Configuring Traffic Control

Set the following objects to control Network Performance Monitoring impact on the device and local network.

[arrisNpmSetupBgTrafficRateEnable](#)

Set to **enable(1)** to use background traffic rate checking. When enabled, the Touchstone device does not run performance monitoring tests if the upstream or downstream traffic exceed the rates set by the next two objects.

[arrisNpmSetupBgTrafficMaxDownstreamRate](#)

The maximum downstream traffic rate, in Kbps, allowed for Network Performance Monitoring tests to run.

[arrisNpmSetupBgTrafficMaxUpstreamRate](#)

The maximum upstream traffic rate, in Kbps, allowed for Network Performance Monitoring tests to run.

Traffic rate checking uses the [ifHCInOctets](#) and [ifHCOctets](#) objects associated with the DOCSIS interface ([ifIndex](#) 2, 3, and 4). If the traffic exceeds the current threshold, the modem:

1. Pauses rate checking for 10 seconds
2. Checks the traffic rate over the next 5 seconds

3. Repeats up to 5 times before abandoning the tests

Setting a Group Reference

Each device includes its CM MAC address in the results, for unique identification. By setting the **arrisNpmSetupGroupReference** object, you can associate devices for average and other aggregation.

The group reference is an arbitrary text string. Suggested grouping identifiers include:

- headend
- CMTS card slot
- fiber node

Using a hierarchical group identifier allows aggregation to be highly flexible.

Configuring Tests

Each test has a group of MIB objects used to configure and control the test. The following sections describe how to set up each test.

Configuring the Webpage Download Test

The **arrisNpmWebDITest** objects configure and control this test. The Webpage Download test downloads up to 10 configured web pages, and measures the time taken.

To set the timeout for this test, set the **arrisNpmSetupWebPageDITestTimeout** object to the desired time, in seconds. This timeout applies to each page downloaded, not the entire test.

To configure URLs to download for the test, set the following two objects in the **arrisNpmSetupWebPageDITestTable**:

arrisNpmSetupWebPageDITestConfigUrl

The URL of the webpage to download. Use an index value between **1** and **10**.

arrisNpmSetupWebPageDITestConfigRowStatus

Set to **createAndGo(4)** to add the URL to the table.

Configuring the DNS Latency Test

The DNS Latency test uses the entries in the **arrisNpmSetupWebPageDITestTable**, recording the time required to perform a DNS resolve of each URL in the table.

By default, the DNS Latency test uses the DNS servers assigned to the CM through DHCP. If you want to override the configured DNS servers, set the following objects to the IP address of the desired servers:

- **arrisNpmSetupDnsPrimaryServerIpAddress**
- **arrisNpmSetupDnsSecondaryServerIpAddress**

If you want to run the DNS Latency test alongside the Webpage Download test, set the **arrisNpmSetupDnsTestEnable** object to **enable(1)**. When enabled, the DNS Latency test automatically runs when you initiate a Webpage Download test.

Configuring the Network Latency Test

The Network Latency test requires one or more dedicated UDP Ping servers to be deployed in the MSO network. The **arrisNpmSetupNetLatencyServerTable** designates up to five Ping servers for the Touchstone device to use for the test.

To configure the Ping servers, set the following objects:

arrisNpmSetupNetLatencyConfigServer

The IP address or FQDN of a Ping server.

arrisNpmSetupNetLatencyConfigServerPort

The UDP port where the Ping server listens for pings.

arrisNpmSetupNetLatencyConfigServerRowStatus

Set to **createAndGo(4)**.

Configure the following optional objects, as needed:

arrisNpmSetupNetLatencyTestPingCount

The number of pings to send to each configured Ping server. Valid range: **1** to **10**. Default: **1**.

arrisNpmSetupNetLatencyTestPingInterval

The time, in milliseconds, between each ping. Valid range: **1** to **36000000**. Default: **50**.

arrisNpmSetupNetLatencyTestPingTimeout

The time, in milliseconds, to wait for a response from the Ping server before declaring the request failed. Valid range: **1000** to **6000**. Default: **3000**.

arrisNpmSetupNetLatencyTestRunUnderLoadEnable

Set this object to **enable(1)** to repeatedly run the Network Latency test during the Webpage Download test. This allows testing latency when the Touchstone device has a CM-originated network load applied to the default service flow. When this option is enabled, the Ping interval is 300ms.

Running Tests

Network Performance tests can be run separately or in a group. The Webpage Download test is the “anchor” test for running multiple tests.

Running the Webpage Download Test

To run the Webpage Download test, write a numeric value to the **arrisNpmSetupWebPageDlTestRunTime** object. The value is a countdown timer, in seconds, used to delay the start of the test. Write a value of **0** to start the test immediately.



Note: Wait at least five minutes for the test to complete before checking the results.

Running the DNS Latency Test

You can run the DNS Latency test by itself, or as part of the Webpage Download test:

- To run the test with the Webpage Download test, set the **arrisNpmSetupDnsTestEnable** object to **enable(1)**.
- To run the DNS Latency test by itself, write to the **arrisNpmSetupDnsTestRunTime** object. The value is a countdown timer, in seconds, used to delay the start of the test. Write a value of **0** to start the test immediately.



Note 1: Wait at least five minutes for the test to complete before checking the results.

Note 2: If the **arrisNpmSetupDnsTestEnable** object is enabled, writing to the **arrisNpmSetupDnsTestRunTime** object has no effect.

Running the Network Latency Test

You can run the Network Latency test by itself, or as part of the Webpage Download test:

- To run the test with the Webpage Download test, set the **arrisNpmSetupNetLatencyTestRunUnderLoadEnable** object to **enable(1)**. Note that when this option is enabled, the ping interval is always 300ms.
- To run the Network Latency test by itself, write to the **arrisNpmSetupNetLatencyTestRunTime** object. The value is a countdown timer, in seconds, used to delay the start of the test. Write a value of **0** to start the test immediately.



Note 1: Wait at least five minutes for the test to complete before checking the results.

Note 2: If the **arrisNpmSetupNetLatencyTestRunUnderLoadEnable** object is enabled, writing to the **arrisNpmSetupNetLatencyTestRunTime** object has no effect.

Results

The Network Performance tests output the results in JSON format. The following is an example of a Webpage Download test for one site:

```
{ 'results': { 'time': '2011-02-08 08:49:37', 'cm': '00ca.1231.3939',
  'group': 'CMTS-MD-1-0-2', 'url': 'www.cnn.com', 'bytes': '12023291',
  'duration': '8923', 'run': '1', 'fail': '0' } }
```

Webpage Download Results

The **arrisNpmResultWebPageDlTestTable** contains the results of the Webpage Download test. The table contains a **arrisNpmResultWebPageDlTestResult** entry for each URL configured in the **arrisNpmSetupWebPageDlTestTable**. The fields in each entry are:

time

The time the test was run, in the format **YYYY-MM-DD HH:MM:SS**.

cm

The CM MAC address.

group

The group string assigned to the device.

url

The URL downloaded.

bytes

The number of bytes transferred.

duration

The time, in milliseconds, elapsed during the test.

run

Returns **1** if the test was run. If **0**, the test was not allowed to run due to background traffic rates.

fail

Returns **0** if the test succeeded, or **1** if the test failed.

DNS Latency Results

The [**arrisNpmResultDnsTestTable**](#) contains the results of the DNS Latency test. The table contains an [**arrisNpmResultDnsTestResult**](#) entry for each URL configured in the [**arrisNpmSetupWebPageDITestTable**](#). The fields in each entry are:

time

The time the test was run, in the format **YYYY-MM-DD HH:MM:SS**.

cm

The CM MAC address.

group

The group string assigned to the device.

server

The IP address of the DNS server used.

url

The URL looked up or downloaded. The number of bytes transferred.

duration

The time, in milliseconds, elapsed during the test.

run

Returns **1** if the test was run. If **0**, the test was not allowed to run due to background traffic rates.

fail

Returns **0** if the test succeeded, or **1** if the test failed.

Network Latency Results

The [arrisNpmResultNetLatencyTestTable](#) contains the results of the Network Latency test. The table contains an [arrisNpmResultNetLatencyTestResult](#) entry for each Ping server configured in the [arrisNpmSetupWebPageDItestTable](#). The fields in each entry are:

time

The time the test was run, in the format **YYYY-MM-DD HH:MM:SS**.

cm

The CM MAC address.

group

The group string assigned to the device.

server

The IP address of the DNS server used.

avg

The average round-trip time (RTT), in milliseconds, measured during the test.

min

The minimum RTT, in milliseconds, measured during the test.

max

The maximum RTT, in milliseconds, measured during the test.

median

The median value of all RTT measurements, in milliseconds.

range

The difference, in milliseconds, between the maximum and minimum RTT values.

std

The standard deviation, in milliseconds, of the sample.

run

The number of Pings sent during the test.

fail

The number of Ping timeouts or DNS failures during the test.

Maintenance

Maintenance involves upgrades, enabling new features, diagnostics, and troubleshooting.

Overview of Maintenance Interfaces

This release of Touchstone firmware uses SNMP for maintenance and troubleshooting.



Note: SNMP access from LAN interfaces is disabled by default. To allow SNMP access to a test device after ranging and registering, set TLV-55 in the CM configuration file.

WebGUI Access Levels and Defaults

The eRouter WebGUI can be accessed from the WAN or LAN interfaces.

Subscriber access is restricted to user-controlled settings, and is only available at LAN interfaces.

- User name: **admin**
- Default password: **password** (can be changed by the subscriber)

From the WAN interface, there are two levels of access available.

- **cusadmin** (limited access for first-tier support)
- **mso** (full access)

Both use the Password of the Day (PWoD) mechanism.

LED Patterns





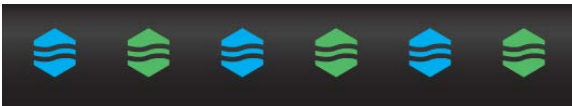
The Touchstone Telephony Modem has eight indicator lights to assist in troubleshooting. Note that not all models have a Battery light.

Wiring Problems Indication

If the Telephony Modem flashes all its lights for more than 10 seconds, this indicates a problem with the telephone wiring—the tip and ring (red and green) wires may be shorted (touching), or there may be undesired voltage on the lines. If this pattern persists for more than 10 seconds, disconnect the telephone lines from the Telephony Modem, then call a wiring technician for assistance.





TM3402 Normal Operation



The following table shows light patterns for the Online LED during normal operation.

Event	Online LED Status	LED Behavior
Power On	Blue (steady)	
Acquiring downstream lock	Blue (slow blink)	
Acquiring upstream lock	Blue (fast blink)	
IP Link/Online	Green (steady)	
Firmware download	Alternating blue and green	

Caution: If all LEDs are blinking blue (fast blink), a grounding defect has been detected. Unplug the device until the grounding defect can be addressed.


The following table shows light patterns for the Wi-Fi LED during normal operation.

Event	Wi-Fi LED Status	LED Behavior
No clients found (5 GHz or dual-band)	Green (slow blink)	
Clients found, but no traffic (5 GHz or dual-band)	Green (steady)	
Clients with active traffic/WPS (5 GHz or dual-band)	Green (fast blink)	
No clients found (2.4 GHz band only)	Blue (slow blink)	

Event	Wi-Fi LED Status	LED Behavior
Clients found, but no traffic (2.4 GHz band only)	Blue (steady)	
Clients with active traffic/WPS (2.4 GHz band only)	Blue (fast blink)	

Caution: If all LEDs are blinking blue (fast blink), a grounding defect has been detected. Unplug the device until the grounding defect can be addressed.

TG3442/TG3452 only: The following table shows light patterns for the Voice LED during normal operation.

Event	Voice LED Status	LED Behavior
All lines on-hook, good battery, AC power present	Green (steady)	
All lines on-hook, good battery, AC power missing	Blue (steady)	
All lines on-hook, low or bad battery	Red (steady)	
Any line off-hook, good battery, AC power present	Green (slow blink)	
Any line off-hook, good battery, AC power missing	Blue (slow blink)	
Any line off-hook, low or bad battery	Red (slow blink)	
Registration in process	Green (slow blink)	
HD Call in process on line 1	Green (fast blink)	
Home alarm triggered	Alternating green and blue	



Caution: If all LEDs are blinking blue (fast blink), a grounding defect has been detected. Unplug the device until the grounding defect can be addressed.



Loopback Testing

Touchstone NCS firmware supports both NETWLOOP and NETWTEST loopback connection modes. You must activate loopback mode on a line from the call server.

Reset to Factory Defaults

Touchstone firmware provides the ability to reset a Telephony Modem to its factory-default configuration. Use one of the following methods to reset a Telephony Modem:

- Press and hold the **Reset** button on the back of the unit for 15 seconds. This resets the E-UE and (for Telephony Gateways) the router to factory defaults.
- Using an SNMP manager, set the **rdkbRgDeviceFactoryReset** object to **1**. This method does not affect router settings.

Using the Password of the Day Tool

This procedure describes the purpose and usage of the ARRIS Password of the Day (PWoD) tool.

The PWoD tool generates the appropriate password to access troubleshooting interfaces.



Note: the PWoD functionality is disabled until a seed is configured, as described in "see *Changing the Seed* (page 264)." When setting up SNMP security, avoid using the default community strings ("Public" and "Private") or open access (no community string).

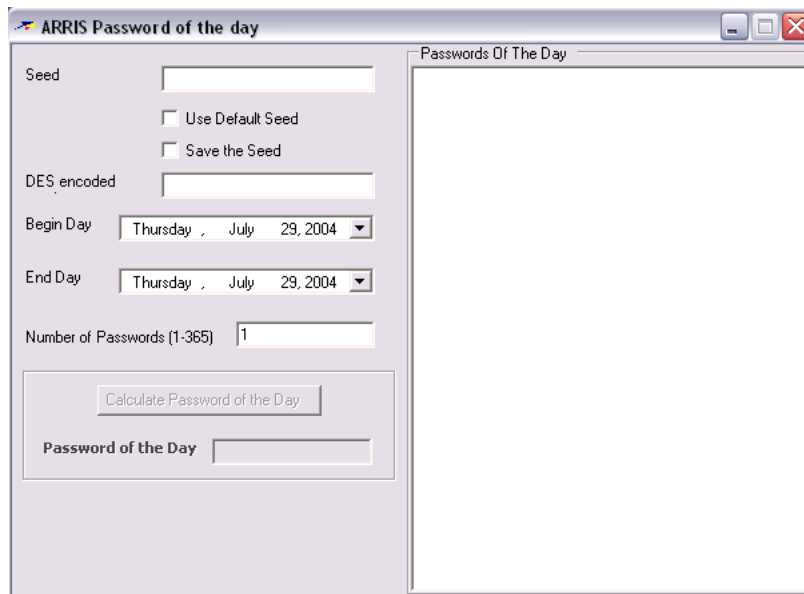
About the Password of the Day Tool

The PWoD tool, **ARRISpwod.exe**, is available through the Firmware Download Center. It is a Windows application, with the following requirements:

- Windows 98, Windows NT, Windows 2000, Windows XP or Windows 7
- Microsoft.Net plus Service Pack 2. The installer provides this package if required.
- Internet Explorer version 5.1 or newer. The newest version of Internet Explorer is available at *the Microsoft web site* (<http://www.microsoft.com/windows/ie/default.asp>).

The PWoD tool can create a single password, or a list of passwords for a range of days (up to 365). For added security, you can also define a seed value used to generate the passwords.

Before performing any of the tasks in this procedure, double-click the file to start the PWoD tool. The following diagram shows the tool.



Action

Perform the following tasks as needed.

Changing the Seed

Follow these steps to change the seed.

1. Start the PWoD tool, if you have not done so already.
2. Enter a new seed value (4 to 8 ASCII characters) in the **Seed** field at the top of the PWoD window. Make sure the **Use Default Seed** box is not checked.

The encoded seed appears in the **DES encoded** field.

Example seed values:

```
abCd1234
Abc#$^&
*!() 1_?
A1h53
abcdEFGH
```

3. If you want to save the seed, make sure the **Save Seed** box is checked.

The PWoD tool saves the new seed to a file called **password.dat**, in the directory where the PWoD is located. Make sure that any computer that can access a password file is reasonably secure.

Touchstone E-UEs also need to have the changed seed so that their internal PWoD generators remain in sync with the external tool. Write the DES encoded seed to the **arrisCmDoc30AccessClientSeed** object in the eDVA configuration file.



Note: To disable Password of the Day functionality, set the `arrisCmDoc30AccessClientSeed` object to all zeroes.

Generating a Single Password

Follow these steps to generate a single password.

1. Start the PWoD tool, if you have not done so already.
2. Make sure the **Save Seed** box is checked, and the **Use Default Seed** checkbox is not checked.
3. Set the **Begin Day** and **End Day** dates to the same date (the default for both fields is the current date).
4. Click **Calculate Password of the Day**.

The Password of the Day appears in the text box at the right of the PWoD tool window. You can select and copy the password as needed.

Generating a List of Passwords

Follow these steps to generate a list of passwords.

1. Start the PWoD tool, if you have not done so already.
2. Make sure the **Save Seed** box is checked, and the **Use Default Seed** checkbox is not checked.
3. Set the **Begin Day** and **End Day** dates to the range of days that you want to generate passwords for (the default for both fields is the current date).
4. Use **Browse** to specify a file name and location for the output file.
5. Click **Calculate Password of the Day**.

The Password of the Day for the first day appears in the text box at the bottom of the PWoD tool window.

The PWoD tool writes the list of passwords to the specified file. The file contains a list of dates and the associated password for that day.

Using the Spectrum Analyzer

The Spectrum Analyzer scans the 2.4 GHz and 5.0 GHz Wi-Fi bands and lists all access points (APs) found. This is equivalent to the AP Scan feature, available on some Touchstone DOCSIS 3.0 devices.

To use the Spectrum Analyzer:

1. Open the web pages for the Gateway.
2. Select the Troubleshooting tab.
3. Select Wi-Fi Spectrum Analyzer. The scan begins immediately.

When the scan completes, the Gateway displays a listing of detected access points similar to the following:

The screenshot shows the ARRIS Gateway web interface. The top navigation bar includes the ARRIS logo, status indicators (N/A, Internet, Wi-Fi, MoCA, Low Security), and user information (Hi mso, Logout). The left sidebar contains a menu with options: Gateway, Connected Devices, Parental Control, Advanced, Troubleshooting (selected), Logs, Diagnostic Tools, Wi-Fi Spectrum Analyzer (highlighted), and Reset/Restore Gateway. The main content area is titled 'Troubleshooting > Wi-Fi Spectrum Analyzer' and features a 'START SCAN' button and a 'SAVE RESULT' button. Below these buttons is a table titled 'Wi-Fi Spectrum Analyzer Data'.

Band	Channel	MAC	SSID	SignalLevel	SignalLevel	Security
2.4GHz	1	F4:0E:83:FA:FD:49	001ac0	-44 dBm	b,g,n	WPA2PSK
		10:86:8C:A8:E0:E0	A8E0E0	-35 dBm	b,g,n	WPAPSKWF
	5	00:1D:D4:D1:F1:F0	TG1672F1-24G	-42 dBm	b,g,n	WPA2PSK
		06:1D:D4:D1:F1:F0	SZ_ATL_1672_Guest24	-42 dBm	b,g,n	WPA2PSK
		10:86:8C:A8:E0:E0	A8E0E0	-57 dBm	b,g,n	WPAPSKWF
		02:1D:D4:D1:F1:F0	ARRIS-F1F2-2	-41 dBm	b,g,n	WPA2PSK
	11	E4:57:40:00:C1:0C	VM8419821	-57 dBm	g,n	WPA2PSK
		FE:51:A4:A3:87:A9	SBG8300Guest-SCOTT	-58 dBm	g,n	WPA2PSK
		10:FE:ED:C1:FF:D3	wemo-demo	-43 dBm	b,g,n	WPAPSKWF
		20:F1:9E:02:8C:C0	-ARRIS-82B2-24G	-57 dBm	b,g,n	WPA2PSK
C0:05:C2:0B:7D:D9		VM6066363	-43 dBm	b,g,n	WPAPSKWF	
5GHz	100	78:23:AE:84:6E:20	Quantenna-wifi0_0	-86 dBm	a,n,ac	WPA2PSK
		9C:1C:12:8E:E0:30	area52	-93 dBm	a,n,ac	WPA2
	124	58:B6:33:24:36:EC	HOSPAC-PRE	-90 dBm	a,n,ac	OPEN
		58:B6:33:64:36:EC	HOSPLB-PRE	-89 dBm	a,n,ac	OPEN
		58:B6:33:A4:36:EC	SL-WIFIPRE	-89 dBm	a,n,ac	OPEN
		58:B6:33:E4:36:EC	TWCWiFi-SL	-90 dBm	a,n,ac	OPEN

The bottom of the interface features a blue footer bar with the text: ARRIS • Customer Support • User Guide • Open Source.

To re-scan, click the START SCAN button. To save the current list, click the SAVE RESULT button.

References

The following reference material is available:

Supported Calling Features

Touchstone firmware supports the following calling features:

- Automatic Number Assignment Confirmation (ANAC) via CID2
- Direct Distance Dialing (DDD)
- Critical Interdigital Timing for Dialing Plan
- International DDD (IDDD) Local Billing Control
- Residence Distinctive Alerting Service
- Free Terminating Service
- Code Restriction & Diversion
- Toll Restricted Service
- DTMF Dialing
- Pulse Dialing
- CLASS™: Calling Number Delivery
- CLASS: Customer Originated Trace
- CLASS: Anonymous Call Rejection
- CLASS: Calling Number Delivery Blocking
- CLASS: Calling Identity Delivery & Suppression
- CLASS: Calling Name Delivery Blocking
- CLASS: Calling Name Delivery
- CLASS: Calling Identity Delivery on Call Waiting
- Speed Calling 8
- Speed Calling 30
- Call Waiting
- Cancel Call Waiting (*70)
- Call Waiting Deluxe
- Access to Telecommunications Relay Service (TDD)
- Intercept Routing for blank/changed/etc. phone numbers
- Customer-Changeable Speed Calling
- Call Forwarding Variable
- Call Forwarding Busy Line
- Call Forwarding — Don't Answer — All Calls
- VIP Alert (Distinctive Ringing)
- Visual Message Waiting Indicator (FSK)
- Message Waiting Tone (stutter dial tone)

- Conference Calling — Six-Way Station Controlled
- Call Hold, Call Pick-up, Toll Free Calling
- Emergency Calling Services (E911)
- Customer Call Back (Automatic Recall) (*69)
- Three-Way Calling
- Service Provider Originated Trace
- Courtesy Ring Generation
- Multiple Directory Numbers on a Line
- Customer Access Treatment (CAT) code restrictions
- Semi-Restricted Originating & Terminating (including 1010xxx blocking)
- Fully Restricted Originating & Terminating
- Single-Digit Dialing
- Manual Line Service
- Direct Connect
- Denied Terminating Service
- Denied Originating Service
- Local Number Portability
- Remote Activation of Call Forwarding (RACF)
- Outside Calling Area Alerting (OCAA)

Country Code Templates

Use the [ppCfgMtaCountryTemplate](#) object to set the country code template.

AR01.1 firmware supports the following country code templates:

Name	Gain settings (dB)		Flash time (ms) (NA load only)	
	Tx	Rx	Min	Max
North America 5/7 (1)	-5	-7	250	1200
Chile (2)	-3	-9.5	40	600
Japan (3)	-4	-8	200	1200
Australia (4)	-3	-9.5	250	1200
Austria (5)	-3	-9.5	85	500
France (6)	-3	-9.5	300	500
Germany (7)	-3	-9.5	300	500
Ireland (8)	-3	-9.5	250	1200
Netherlands (9) (Euro-DOCSIS default)	-3	-9.5	300	500
Portugal (10)	-3	-9.5	100	300

Name	Gain settings (dB)		Flash time (ms) (NA load only)	
	Tx	Rx	Min	Max
Spain (11)	-3	-9.5	90	650
Belgium (12)	-3	-9.5	50	600
Poland (13)	-3	-9.5	50	520
Israel (14)	-2	-4	200	800
Czech Republic (15)	-3	-9.5	250	1200
Brazil (16)	-3	-9.5	220	320
North America 3/3 (17) (DOCSIS default)	-3	-3	250	1200
North America 0/9 (18)	0	-9	250	1200
Netherlands 0/9 (19)	0	-9	300	500
Japan (20) (Japan default)	-4	-8	200	1200
Hungary (21)	-3	-9.5	60	200
Sweden (22)	-3	-9.5	250	1200
Norway (23)	-3	-9.5	90	800
Slovakia (24)	-3	-9.5	250	1200
Japan 600L412 (25)	-4	-12	200	1200
Mexico (26)	0	-7	100	800
Panama (27)	-3	-9.5	220	320
MexicoC (28)	0	-7	100	800
Switzerland (29)	-3	-9.5	88	600
Poland1010 (30)	-13	-19.5	70	250
Germany2 (31)	-3	-9.5	300	500
North America 6/6 (32) (.TW default)	-6	6	250	1200
Argentina (33)	0	-7	100	1100

North American Ring Cadences

The following ring cadences may be provisioned using the PacketCable NCS Signaling MIB (see PKT-SP-MIB-SIG1.5-I01-050128). The following table shows the default ring cadences for North America.

Name	Description	Default
L/RG	Standard Ringing	2 seconds on, 4 seconds off
L/R0	Distinctive Ringing #0	2 seconds on, 4 seconds off
L/R1	Distinctive Ringing #1	2 seconds on, 4 seconds off
L/R2	Distinctive Ringing #2	800 ms on, 400 ms off, 800 ms on, 4 seconds off
L/R3	Distinctive Ringing #3	400 ms on, 200 ms off, 400 ms on, 200 ms off, 800 ms on, 4 seconds off
L/R4	Distinctive Ringing #4	300 ms on, 200 ms off, 1 second on, 200 ms off, 300 ms on, 4 seconds off
L/R5	Distinctive Ringing #5	500 ms on, 5.5 seconds off (not repeated)
L/R6	Distinctive Ringing #6	2 seconds on, 4 seconds off
L/R7	Distinctive Ringing #7	2 seconds on, 4 seconds off
L/RS	Ring Splash	500 ms on, 5.5 seconds off (not repeated)
L/RT	Ringback Tone	2 seconds on, 4 seconds off

Touchstone firmware uses the default ring cadences shown above when the country template is provisioned to be one of the following:

- **northAmerica57**
- **northAmerica33**
- **northAmerica09**
- **northAmerica66**

Template (i.e. hard-coded country-specific) based ring cadences are used by default when the country template is provisioned. This default behavior may be overridden by setting the "Provisioned Ring Cadences" CallP Feature Switch setting, and updating the eDVA configuration file with the provisioning for the appropriate MIB objects to define ring cadences (for example, [pkcSigDevRgCadence](#)). To make this setting, add **0x02000000** to the current feature switch setting in the CM configuration file.

Customizing Default Ring Cadences

Any of the above ring cadences may be customized in the eDVA configuration file. All MIB objects are eDVA based; therefore, the first cadence is index 0.

The ring cadence is internally represented as a 64-bit string and provisioned in hex format. The ring cadence representation starts with the first **1** in the bit string pattern. Leading zeros are ignored, thus shortening the overall ring cadence duration. Each bit represents 100 ms of ringing (or tone in the case of L/RT); **1** is ring on, **0** is ring off.

All 64 bits must be provisioned. The least significant 4 bits are used for representing repeatable characteristics: **0000** indicates that the ring cadence repeats, and **1000** indicates a non-repeatable ring cadence. Therefore, only the first 60 bits are used to represent the actual ring cadence for a maximum duration of 6 seconds.

As mentioned earlier, shorter ring cadences may be provisioned by padding the ring cadence with leading zeros. For example, a ring cadence of 0.5 seconds on, 4 seconds off, repeatable, has a value of **0x0001F00000000000** and would be provisioned in the eDVA configuration file as **00. 01. F0. 00. 00. 00. 00. 00.**

Default Tone Settings

The following tables show default tones for supported country templates. The columns in each table are as follows:

type

The type of tone (busy, dial tone).

level

The tone level, in dB.

Freq. Type

Either **1** (first frequency modified by the second) or **2** (summation).

Freq.

The number of frequencies used to generate the tone (1–4).

Frequencies

The frequencies used to generate the tone.

on/off

The number of on/off cycles in the tone pattern (1–4).

1st tone – 4th tone

The duration, in milliseconds, of the on/off segments of each tone cycle.

rep. count

The number of times the tone pattern is repeated.

tone steady

Which of the four tones, if any, are held indefinitely after the pattern completes (used, for example, with stutter dial).



Note: Not all firmware loads support all country code templates and tones.

North America

type	level	Freq. Type	# Freq.	Frequencies				tone number	on time	off time	rep. count	tone steady
				1st	2nd	3rd	4th					
Busy(16)	-45.3	1	2	480	620	0	0	1	500	500	30	2
Confirmation(17)	-13	1	2	350	440	0	0	1	1000	1000	0	4
								2	1000	1000		
								3	1000	1000		
Dial(18)	-13	1	2	350	440	0	0	1	5000	0	0	1
Offhook Warning (20)	-6	1	4	1400	2060	2450	2600	1	100	100	5000	2
Ringback(21)	-19	1	2	440	480	0	0	1	2000	4000	5000	2
Reorder(22)	-24	1	2	480	620	0	0	1	250	250	60	2
Stutterdial(23)	-13	1	2	350	440	0	0	1	100	100	0	4
								2	100	100		
								3	100	0		
								4	5000	0		
Message Waiting (24)	-13	1	2	350	440	0	0	1	100	100	10	2
								2	5000	0		
Call Waiting 1 (25)	-13	1	1	440	0	0	0	1	300	0	1	2
Special Information Tone(30)	-17	1	3	950	1400	1800	0	1	330	1000	2	3

CableLabs Wi-Fi Objects MIB

The CableLabs Wi-Fi MIB contains management objects for the Wi-Fi interface. The objects in this MIB map to TR-069 objects.

clabWIFIWiFiRadioNumberOfEntries

(read-only) The number of entries in the **clabWIFIRadioTable**. Equivalent to the TR-181 Device.WiFi.RadioNumberOfEntries object.

clabWIFIWiFiSSIDNumberOfEntries

(read-only) The number of entries in the **clabWIFISSIDTable**. Equivalent to the TR-181 Device.WiFi.SSIDNumberOfEntries object.

clabWIFIWiFiAccessPointNumberOfEntries

(read-only) The number of entries in the Access Point table. Equivalent to the TR-181 Device.WiFi.AccessPointNumberOfEntries object.

clabWIFIWiFiEndPointNumberOfEntries

(read-only) Equivalent to the TR-181 Device.WiFi.EndPointNumberOfEntries object.

clabWIFIRadioTable

Equivalent to the TR-181 Device.WiFi.Radio group. The table uses the **ifIndex** of each radio as an index. The following entries are available:

clabWIFIRadioEnable

Enables or disables the radio that this table entry represents. Equivalent to the TR-181 Device.WiFi.Radio.{i}.Enable object.

clabWIFIRadioStatus

The radio status. Equivalent to the TR-181 Device.WiFi.Radio.{i}.Status object. The status is one of the following:

- **up**(1)
- **down**(2)
- **unknown**(4)
- **dormant**(5)
- **notPresent**(6)
- **lowerLayerDown**(7)
- **error**(8)

clabWIFIRadioAlias

A handle, used by the ACS to refer to this entry. Equivalent to the TR-181 Device.WiFi.Radio.{i}.Alias object.

clabWIFIRadioName

Equivalent to the TR-181 Device.WiFi.Radio.{i}.Name object.

clabWIFIRadioLastChange

The time, in seconds, since the last change to the radio configuration. Equivalent to the TR-181 Device.WiFi.Radio.{i}.LastChange object.

clabWIFIRadioLowerLayers

Equivalent to the TR-181 Device.WiFi.Radio.{i}.LowerLayers object.

clabWIFIRadioUpstream

Enables or disables the upstream radio connection. Equivalent to the TR-181 Device.WiFi.Radio.{i}.Upstream object.

clabWIFIRadioMaxBitRate

The maximum bitrate, in Mbps. Equivalent to the TR-181 Device.WiFi.Radio.{i}.MaxBitRate object.

clabWIFIRadioSupportedFrequencyBands

Equivalent to the TR-181 Device.WiFi.RadioSupportedFrequencyBands object.

clabWIFIRadioOperatingFrequencyBand

The operating frequency band, one of **n2dot4Ghz**(1) or **n5Ghz**(2). Equivalent to the TR-181 Device.WiFi.Radio.{i}.OperatingFrequencyBand object.

clabWIFIRadioSupportedStandards

Equivalent to the TR-181 Device.WiFi.Radio.{i}.SupportedStandards object.

clabWIFIRadioOperatingStandards

The 802.11 standard for this radio, one of: **a**(1), **b**(2), **g**(3), or **n**(5). Equivalent to the TR-181 Device.WiFi.Radio.{i}.OperatingStandards object.

clabWIFIRadioPossibleChannels

A list of available channels. Equivalent to the TR-181 Device.WiFi.Radio.{i}.PossibleChannels object.

clabWIFIRadioChannelsInUse

A list of channels in use. Equivalent to the TR-181 Device.WiFi.Radio.{i}.ChannelsInUse object.

clabWIFIRadioChannel

The current channel in use. Equivalent to the TR-181 Device.WiFi.Radio.{i}.Channel object.

clabWIFIRadioAutoChannelSupported

(read-only) Returns **true**(1) if this radio supports automatic channel selection. Equivalent to the TR-181 Device.WiFi.Radio.{i}.AutoChannelSupported object.

clabWIFIRadioAutoChannelEnable

Set to **true**(1) to enable automatic channel selection on this radio. Equivalent to the TR-181 Device.WiFi.Radio.{i}.AutoChannelEnable object.

clabWIFIRadioAutoChannelRefreshPeriod

The refresh period, in seconds. Equivalent to the TR-181 Device.WiFi.Radio.{i}.AutoChannelRefreshPeriod object.

clabWIFIRadioOperatingChannelBandwidth

The bandwidth of the channel in use, one of: **n20Hhz**(1), **n40Mhz**(2), or **auto**(3) (the default). Equivalent to the TR-181 Device.WiFi.Radio.{i}.OperatingChannelBandwidth object.

clabWIFIRadioExtensionChannel

The location of the extension channel, one of: **aboveControl Channel** (1), **belowControl Channel** (2), or **auto**(3) (the default). Equivalent to the TR-181 Device.WiFi.Radio.{i}.ExtensionChannel object.

clabWIFIRadioGuardInterval

The guard interval, one of: **n400nsec**(1), **n800nsec**(2), or **auto**(3) (the default). Equivalent to the TR-181 Device.WiFi.Radio.{i}.RadioGuardInterval object.

clabWIFIRadioMCS

The MCS configuration for this radio. Valid range: - **1** to **31**. Equivalent to the TR-181 Device.WiFi.Radio.{i}.RadioMCS object.

clabWIFIRadioTransmitPowerSupported

Equivalent to the TR-181 Device.WiFi.Radio.{i}.TransmitPowerSupported object.

clabWIFIRadioTransmitPower

The transmit power, as a percentage of the maximum power. Valid range: **1** to **100**. Equivalent to the TR-181 Device.WiFi.Radio.{i}.TransmitPower object.

clabWIFIRadioIEEE80211hSupported

(read-only) Returns **true**(1) if the radio supports 802.11h. Equivalent to the TR-181 Device.WiFi.Radio.{i}.IEEE80211hSupported object.

clabWIFIRadioIEEE80211hEnabled

Set to **true**(1) to enable 802.11h support on this radio. Equivalent to the TR-181 Device.WiFi.Radio.{i}.IEEE80211hEnabled object.

clabWIFIRadioRegulatoryDomain

Equivalent to the TR-181 Device.WiFi.Radio.{i}.RegulatoryDomain object.

clabWIFIRadioStatsTable

A table of statistics for each radio. The table is indexed by the **ifIndex** for each radio. Equivalent to the TR-181 Device.WiFi.Radio.{i}.Stats group.



Note: all objects in this table are read-only.

clabWIFIRadioStatsBytesSent

The number of bytes sent by this radio. Equivalent to the TR-181 Device.WiFi.Radio.{i}.Stats.BytesSent object.

clabWIFIRadioStatsBytesReceived

The number of bytes received by this radio. Equivalent to the TR-181 Device.WiFi.Radio.{i}.Stats.BytesReceived object.

clabWIFIRadioStatsPacketsSent

The number of packets sent by this radio. Equivalent to the TR-181 Device.WiFi.Radio.{i}.Stats.PacketsSent object.

clabWIFIRadioStatsPacketsReceived

The number of packets received by this radio. Equivalent to the TR-181 Device.WiFi.Radio.{i}.Stats.PacketsReceived object.

clabWIFIRadioStatsErrorsSent

The number of errored packets sent by this radio. Equivalent to the TR-181 Device.WiFi.Radio.{i}.Stats.ErrorsSent object.

clabWIFIRadioStatsErrorsReceived

The number of errored packets received by this radio. Equivalent to the TR-181 Device.WiFi.Radio.{i}.Stats.ErrorsReceived object.

clabWIFIRadioStatsDiscardPacketsSent

The number of discarded packets sent by this radio. Equivalent to the TR-181 Device.WiFi.Radio.{i}.Stats.DiscardPacketsSent object.

clabWIFIRadioStatsDiscardPacketsReceived

The number of discarded packets received by this radio. Equivalent to the TR-181 Device.WiFi.Radio.{i}.Stats.DiscardPacketsReceived object.

clabWIFISSIDTable

A table of available SSIDs, indexed by the **ifIndex** of each available SSID. Equivalent to the TR-181 Device.WiFi.SSID{i} group.

clabWIFISSIDEnable

Set to **true**(1) to enable this SSID. Equivalent to the TR-181 Device.WiFi.SSID{i}.Enable object.

clabWIFISSIDStatus

(read-only) The SSID status, one of: **up**(1), **down**(2), **unknown**(4), **dormant**(5), **notPresent**(6), **lowerLayerDown**(7), or **error**(8). Equivalent to the TR-181 Device.WiFi.SSID{i}.Status object.

clabWIFISSIDAlias

A handle, used by the ACS to refer to this entry. Equivalent to the TR-181 Device.WiFi.SSID{i}.Alias object.

clabWIFISSIDName

(read-only) The SSID name. Equivalent to the TR-181 Device.WiFi.SSID{i}.Name object.

clabWIFISSIDLastChange

(read-only) The time, in seconds, since the last change to this SSID. Equivalent to the TR-181 Device.WiFi.SSID{i}.LastChange object.

clabWIFISSIDLowerLayers

Equivalent to the TR-181 Device.WiFi.SSID{i}.LowerLayers object.

clabWIFISSIDBSSID

(read-only) Equivalent to the TR-181 Device.WiFi.SSID{i}.BSSID object.

clabWIFISSIDMACAddress

(read-only) The MAC address of the SSID. Equivalent to the TR-181 Device.WiFi.SSID{i}.MACAddress object.

clabWIFISSIDSSID

Equivalent to the TR-181 Device.WiFi.SSID{i}.SSID object.

clabWIFISSIDRowStatus

The RowStatus of this table entry.

clabWIFISSIDStatsTable

A table of statistics for each SSID, indexed by the **ifIndex** of each SSID. Equivalent to the TR-181 Device.WiFi.SSID{i}.Stats group.



Note: all objects in this table are read-only.

clabWIFISSIDStatsBytesSent

The number of bytes sent by the SSID. Equivalent to the TR-181 Device.WiFi.SSID{i}.Stats.BytesSent object.

clabWIFISSIDStatsBytesReceived

The number of bytes received by the SSID. Equivalent to the TR-181 Device.WiFi.SSID{i}.Stats.BytesReceived object.

clabWIFISSIDStatsPacketsSent

The number of bytes sent by the SSID. Equivalent to the TR-181 Device.WiFi.SSID{i}.Stats.PacketsSent object.

clabWIFISSIDStatsPacketsReceived

The number of packets received by the SSID. Equivalent to the TR-181 Device.WiFi.SSID{i}.Stats.PacketsReceived object.

clabWIFISSIDStatsErrorsSent

The number of errored packets sent by the SSID. Equivalent to the TR-181 Device.WiFi.SSID{i}.Stats.ErrorsSent object.

clabWIFISSIDStatsErrorsReceived

The number of errored packets received by the SSID. Equivalent to the TR-181 Device.WiFi.SSID{i}.Stats.ErrorsReceived object.

clabWIFISSIDStatsUnicastPacketsSent

The number of unicast packets sent by the SSID. Equivalent to the TR-181 Device.WiFi.SSID{i}.Stats.UnicastPacketsSent object.

clabWIFISSIDStatsUnicastPacketsReceived

The number of unicast packets received by the SSID. Equivalent to the TR-181 Device.WiFi.SSID{i}.Stats.UnicastPacketsReceived object.

clabWIFISSIDStatsDiscardPacketsSent

The number of discarded packets sent by the SSID. Equivalent to the TR-181 Device.WiFi.SSID{i}.Stats.DiscardPacketsSent object.

clabWIFISSIDStatsDiscardPacketsReceived

The number of discarded packets received by the SSID. Equivalent to the TR-181 Device.WiFi.SSID{i}.Stats.DiscardPacketsReceived object.

clabWIFISSIDStatsMulticastPacketsSent

The number of multicast packets sent by the SSID. Equivalent to the TR-181 Device.WiFi.SSID{i}.Stats.MulticastPacketsSent object.

clabWIFISSIDStatsMulticastPacketsReceived

The number of multicast packets received by the SSID. Equivalent to the TR-181 Device.WiFi.SSID{i}.Stats.MulticastPacketsReceived object.

clabWIFISSIDStatsBroadcastPacketsSent

The number of broadcast packets sent by the SSID. Equivalent to the TR-181 Device.WiFi.SSID{i}.Stats.BroadcastPacketsSent object.

clabWIFISSIDStatsBroadcastPacketsReceived

The number of broadcast packets received by the SSID. Equivalent to the TR-181 Device.WiFi.SSID{i}.Stats.BroadcastPacketsReceived object.

clabWIFISSIDStatsUnknownProtoPacketsReceived

The number of unknown protocol packets received by the SSID. Equivalent to the TR-181 Device.WiFi.SSID{i}.Stats.UnknownProtoPacketsReceived object.

clabWIFIAccessPointTable

A table of known access points. Equivalent to the TR-181 Device.WiFi.AccessPoint{i} group.

clabWIFIAccessPointEnable

Set to **true**(1) to enable this access point. Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.Enable object.

clabWIFIAccessPointStatus

(read-only) The access point status, one of: **disabled**(1), **enabled**(2), **misconfigured**(3), or **error**(4). Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.Status object.

clabWIFIAccessPointAlias

A handle, used by the ACS to refer to this entry. Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.Alias object.

clabWIFIAccessPointSSIDReference

The path name to a row in the Device.WiFi.SSID table. Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.Reference object.

clabWIFIAccessPointSSIDAdvertisementEnabled

True if beacons advertise the SSID name. Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.AdvertisementEnabled object.

clabWIFIAccessPointRetryLimit

The maximum number of retransmissions for a packet. Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.RetryLimit object.

clabWiFiAccessPointWMMCapability

(read-only) True if this access point supports Wi-Fi Multimedia (WMM) Access Categories. Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.WMMCapability object.

clabWiFiAccessPointUAPSDCapability

(read-only) True if this access point supports WMM Unscheduled Automatic Power Save Delivery (UAPSD). Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.UAPSDCapability object.

clabWiFiAccessPointWMMEnable

Set to **true**(1) to enable WMM for this access point. Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.WMMEnable object.

clabWiFiAccessPointUAPSEnable

True if this access point supports WMM Unscheduled Automatic Power Save Delivery (UAPSD). Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.UAPSEnable object.

clabWiFiAccessPointAssociatedDeviceNumberOfEntries

(read-only) The number of entries in the **clabWiFiAssociatedDeviceTable**. Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.AssociatedDeviceNumberOfEntries object.

clabWiFiAccessPointRowStatus

The RowStatus of this table entry.

clabWiFiAccessPointSecurityTable

Provides security-related parameters that apply to a device acting as an Access Point. Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.Security group. This table is indexed by the access point ID.

clabWiFiAccessPointSecurityModesSupported

(read-only) A comma-separated list of strings, indicating the supported security modes, one of: **none**(1), **wep64**(2), **wep128**(3), **wpaPersonal** (4), **wpa2Personal** (5), **wpawpa2Personal** (6), **wpaEnterprise**(7), **wpa2Enterprise**(8), or **wpawpa2Enterprise**(9). Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.Security.ModesSupported object.

clabWiFiAccessPointSecurityModeEnabled

The security mode in use, one of: **none**(1), **wep64**(2), **wep128**(3), **wpaPersonal** (4), **wpa2Personal** (5), **wpawpa2Personal** (6), **wpaEnterprise**(7), **wpa2Enterprise**(8), or **wpawpa2Enterprise**(9). Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.Security.ModeEnabled object.

clabWiFiAccessPointSecurityWEPKey

The WEP key, for access points using WEP. Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.Security.WEPKey object.

clabWiFiAccessPointSecurityPreSharedKey

The pre-shared key. Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.Security.PreSharedKey object.

clabWiFiAccessPointSecurityKeyPassphrase

The key passphrase. Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.Security.KeyPassphrase object.

clabWiFiAccessPointSecurityRekeyingInterval

The rekeying interval, in seconds. Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.Security.RekeyingInterval object.

clabWiFiAccessPointSecurityRadiusServerIPAddrType

Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.Security.RadiusServerIPAddrType object.

clabWiFiAccessPointSecurityRadiusServerIPAddr

The IP address of the RADIUS server used for WLAN security. Applies only to WPA or WPA2 Enterprise modes. Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.Security.RadiusServerIPAddr object.

clabWiFiAccessPointSecurityRadiusServerPort

The port number of the RADIUS server. Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.Security.RadiusServerPort object.

clabWiFiAccessPointSecurityRadiusSecret

The secret used for handshaking with the RADIUS server. Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.Security.RadiusSecret object.

clabWiFiAccessPointSecurityRowstatus

The RowStatus of this table entry.

clabWiFiAccessPointWPSTable

The WPS configuration table, indexed by access point ID. Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.WPS group.

clabWiFiAccessPointWPSEnable

Set to **true**(1) to enable WPS for this access point. Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.WPS.Enable object.

clabWiFiAccessPointWPSConfigMethodsSupported

(read-only) A comma-separated list of strings, indicating supported WPS configuration methods one of: **usbFlashDrive**(1), **ethernet**(2), **externalNFCToken**(3), **integratedNFCToken**(4), **nfcInterface**(5), **pin**(7), or **pushButton**(8). Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.WPS.ConfigMethodsSupported object.

clabWiFiAccessPointWPSConfigMethodsEnabled

The WPS configuration methods enabled, one of: **usbFlashDrive**(1), **ethernet**(2), **externalNFCToken**(3), **integratedNFCToken**(4), **nfcInterface**(5), **pin**(7), or **pushButton**(8). Equivalent to the TR-181 Device.WiFi.AccessPoint{i}.WPS.ConfigMethodsEnabled object.

clabWiFiAccessPointWPSRowStatus

The RowStatus for this table entry.

clabWiFiAssociatedDeviceTable

A table of devices associated with an access point, indexed by access point ID and device number. Equivalent to the TR-181 Device.WiFi.AssociatedDevice{i} object.



Note: all objects in this table are read-only.

clabWiFiAssociatedDeviceMACAddress

The MAC address of the associated device. Equivalent to the TR-181 Device.WiFi.AssociatedDevice{i}.MACAddress object.

clabWiFiAssociatedDeviceAuthenticationState

True if the associated device has authenticated. Equivalent to the TR-181 Device.WiFi.AssociatedDevice{i}.AuthenticationState object.

clabWiFiAssociatedDeviceLastDataDownlinkRate

The last measured downlink data rate to the device, in kbps. Equivalent to the TR-181 Device.WiFi.AssociatedDevice{i}.LastDataDownlinkRate object.

clabWiFiAssociatedDeviceLastDataUplinkRate

The last measured downlink data rate to the device, in kbps. Equivalent to the TR-181 Device.WiFi.AssociatedDevice{i}.DeviceLastDataUplinkRate object.

clabWiFiAssociatedDeviceSignalStrength

The measured signal strength from the device, in dBm. Equivalent to the TR-181 Device.WiFi.AssociatedDevice{i}.SignalStrength object.

clabWiFiAssociatedDeviceRetransmissions

The number of retransmitted packets to the device. Equivalent to the TR-181 Device.WiFi.AssociatedDevice{i}.Retransmissions object.

clabWiFiAssociatedDeviceActive

Returns **true**(1) if the device is active. Equivalent to the TR-181 Device.WiFi.AssociatedDevice{i}.Active object.

clabWiFiDataRateStatsTable

A table of statistics for each speed rate of an 802.11 LAN interface object. The table is indexed by the ifIndex of each radio, and by data rate (in Mbps).



Note: all objects in this table are read-only.

clabWiFiDataRateStatsFramesSent

The total number of frames transmitted from the interface (not marked as duplicated). The value of this counter may be reset to zero when the Gateway is rebooted.

clabWiFiDataRateStatsFramesRetransmissionsSent

The total number of frames retransmitted from the interface (marked as duplicated). The value of this counter may be reset to zero when the Gateway is rebooted.

clabWIFIDataRateStatsFramesReceived

The total number of frames received on this interface (not marked as duplicated).
The value of this counter may be reset to zero when the Gateway is rebooted.

clabWIFIDataRateStatsFramesDuplicatedReceived

The total number of duplicated frames received on this interface. The value of this counter may be reset to zero when the Gateway is rebooted.

clabWIFIPeriodicStatsTable

This table contains periodic statistics for an 802.11 SSID on a CPE device. Note that these statistics refer to the link layer, not to the physical layer.

The table is indexed by SSID, the statistics interval, and finally the statistics ID.



Note: all objects in this table are read-only.

clabWIFIPeriodicStatsDeviceMACAddress

The MAC address of the device associated with these statistics.

clabWIFIPeriodicStatsFramesSent

The total number of frames transmitted out of the interface. For 802.11a/b/g, this counter corresponds to the total MSDUs (MAC Service Data Unit) transmitted. For 802.11n or 802.11ac, this counter corresponds to the A-MSDU (Aggregation MSDU).

clabWIFIPeriodicStatsDataFramesSentAck

The total number of MSDU frames marked as duplicates and non-duplicates acknowledged.

clabWIFIPeriodicStatsDataFramesSentNoAck

The total number of MSDU frames retransmitted out of the interface (i.e., marked as duplicate and non-duplicate) and not acknowledged. This count does not include lost frames.

clabWIFIPeriodicStatsDataFramesLost

The total number of MSDU frames retransmitted out of the interface that were not acknowledged and discarded for reaching max number of retransmissions.

clabWIFIPeriodicStatsFramesReceived

The total number of frames received by the interface. For 802.11a/b/g, this counter corresponds to the total MSDUs (MAC Service Data Unit) received. For 802.11n or 802.11ac, this counter corresponds to the A-MSDU (Aggregation MSDU) and MSDUs.

clabWIFIPeriodicStatsDataFramesReceived

The total number of MSDU frames received and marked as non-duplicates.

clabWIFIPeriodicStatsDataFramesDuplicateReceived

The total number of duplicated frames received on this interface.

clabWIFIPeriodicStatsProbesReceived

The total number of probes received by this interface.

clabWIFIPeriodicStatsProbesRejected

The total number of probes rejected by this interface.

clabWIFIPeriodicStatsRSSI

The signal strength, in dBm, observed at the antenna receiver for a current transmission object.

clabWIFIPeriodicStatsSNR

The Signal-to-Noise ratio (SNR), in dB, at the receiver.

clabWIFIPeriodicStatsDisassociations

The total number of client disassociations.

clabWIFIPeriodicStatsAuthenticationFailures

The total number of authentication failures.

clabWIFIPeriodicStatsLastTimeAssociation

The last time, in date/time format, this device was associated.

clabWIFIPeriodicStatsLastTimeDisassociation

The last time, in date/time format, the client disassociated from the interface. A value of all zeros indicates the client is currently associated.

clabWIFISSIDPolicyTable

This table defines the configuration of policies, behaviors and event thresholds controlled per SSID object.

clabWIFISSIDPolicyBlockAfterAttempts

The maximum number of attempts a client is allowed to attempt registration before being denied access. Exceeding this value generates one event. Events from the same client should not reoccur more than once an hour. The default value of **0** indicates no restriction on attempts.

clabWIFISSIDPolicyAllocatedBandwidth

The the maximum bandwidth, in Mbps, reserved for a particular interface. The default value of **0** indicates no limit.

clabWIFISSIDPolicyAuthenticationFailures

The number of authentication failures a station simultaneously produces to generate the event. Events from same client should not reoccur more than once an hour. The default value of **0** indicates no threshold and events of this type are not generated.

clabWIFISSIDPolicyNonAuthenticatedTraffic

The number of non-authenticated messages received from a station to generate an event. Events from same client should not reoccur more than once an hour. The default value of **0** indicates no threshold, and events of this type are not generated.

clabWIFISSIDPolicyAssociationFailures

The number of simultaneous association failures from a station to generate an event. Events from same client should not reoccur more than once an hour. The default value of **0** indicates no threshold, and events of this type are not generated.

clabWIFISSIDPolicyStatsInterval

The interval, in minutes, to collect per-interval statistics. The default value of **0** indicates no interval and values reported are snapshots at the time of the request.

clabWIFISSIDPolicySNRThreshold

The threshold, in dB, to report SNR. The default value of - **100** indicates no threshold, and events of this type are not generated.

clabWIFISSIDPolicyANPThreshold

The threshold, in dBm, to report the Average Noise plus Interference. The default value of - **100** indicates no threshold, and events of this type are not generated.

clabWIFISSIDPolicyLowReceivedPowerThreshold

The power level threshold, in dBm, to generate an event whenever the station received power is below the threshold. The default value of - **100** indicates no threshold is set, and events of this type are not generated.

clabWIFISSIDPolicyLowPowerDeniedAccessThreshold

The power level threshold, in dBm, to deny client association whenever the station received power is below the threshold. The default value of - **100** indicates no threshold, and events of this type are not generated.

clabWIFISSIDPolicyLowPowerDissassociationThreshold

The threshold, in dBm, to report Disassociation due to low power. The Wi-Fi GW should refuse associations when the power level is below this RSSI level. The default value of - **100** indicates no threshold, and events of this type are not generated.

clabWIFISSIDPolicyRowStatus

The RowStatus of this entry.

clabWIFIClientSessionsTable

Entries in this table represent current and closed sessions (association connections), indexed by access point ID and a session index. When the maximum number of instances is reached, the oldest closed session instance is replaced by a newly created client association object.



Note: all objects in this table are read-only.

clabWIFIClientSessionsDeviceMACAddress

The MAC address of an associated client device.

clabWIFIClientSessionsStart

The time when the session started.

clabWIFIClientSessionsStop

The time when the session ended. If the session is current, this object returns all zeros.

clabWIFIClientSessionsTerminationCode

The Reason Code or Status Code that led to ending the association. Reason code and Status code overlaps. The context of the type of termination is provided by the **clabWIFIClientSessionsTerminationMeaning** object. The value zero indicates the session is active.

clabWIFIClientSessionsTerminationMeaning

The meaning of the Reason Code or Status Code for the ended session. When the session is still active, this object returns an empty string.

clabWIFIClientStatsTable

A table of accumulative statistics for each client station, indexed by access point ID, the statistics interval, and an internal counter. A station is reported only after it is associated for the first time.



Note: all objects in this table are read-only.

clabWIFIClientStatsDeviceMACAddress

The MAC address of an associated client device.

clabWIFIClientStatsFramesSent

The total number of frames transmitted out of the interface. For 802.11a/b/g interfaces, this counter corresponds to the total of MSDUs (MAC Service Data Unit) transmitted. For 802.11n or 802.11ac, this corresponds to the A-MSDU (Aggregation MSDU).

clabWIFIClientStatsDataFramesSentAck

The total number of MSDU frames marked as duplicates and non duplicates acknowledged.

clabWIFIClientStatsDataFramesSentNoAck

The total number of MSDU frames retransmitted out of the interface (i.e., marked as duplicate and non-duplicate) and not acknowledged but not including those defined in dataFramesLost.

clabWIFIClientStatsDataFramesLost

The total number of MSDU frames retransmitted out of the interface that were not acknowledged and discarded for reaching max number of retransmissions.

clabWIFIClientStatsFramesReceived

The total number of frames received by the interface. For 802.11a/b/g interfaces, this counter corresponds to the total of MSDUs (MAC Service Data Unit) received. For 802.11n or 802.11ac, this corresponds to the A-MSDU (Aggregation MSDU).

clabWIFIClientStatsDataFramesReceived

The total number of MSDU frames received and marked as non-duplicates.

clabWIFIClientStatsDataFramesDuplicateReceived

The total number of duplicated frames received on this interface.

clabWIFIClientStatsProbesReceived

The total number of probes received.

clabWIFIClientStatsProbesRejected

The total number of probes rejected.

clabWIFIClientStatsRSSI

The Received Signal Strength Indicator, the energy (in dBm) observed at the antenna receiver for a current transmission.

clabWIFIClientStatsSNR

The Signal-to-Noise ratio (SNR), in dB, received from the access point.

clabWIFIClientStatsDisassociations

The total number of client disassociations.

clabWIFIClientStatsAuthenticationFailures

The total number of authentication failures.

clabWIFIClientStatsLastTimeAssociation

The last time the client was associated.

clabWIFIClientStatsLastTimeDisassociation

The last time the client disassociated from the interface. A value of all zeros value means the client is currently associated.

clabWIFIRadiusClientTable

This table describes Radius clients for the Access Point 802.1x Authenticator for WPA Enterprise, indexed by access point ID.

clabWIFIRadiusClientNAS-Identifier

The Radius attribute NAS-Identifier used in Access request messages. The device always sends the Radius parameter NAS-IP-Address, and sends the NAS-Identifier parameter when this attribute is set to other than the zero-length string. The NAS-Identifier attribute can be used as a hint to indicate the authentication server the SSID domain where the WiFi endpoint tries to authenticate, i.e., when more than one SSID domains are using the same Radius server instance.

clabWIFIRadiusClientLocationPolicy

The string value of the Radius Basic-Location-Policy-Rules attribute per RFC 5580.

clabWIFIRadiusClientOperatorName

The string value of the Radius Operator-Name attribute per RFC 5580.

clabWIFIRadiusClientLocationInformation

The string value of the Radius Location-Information attribute per RFC 5580.

clabWIFIRadiusClientLocationData

The string value of the Radius LocationData attribute per RFC 5580.

clabWIFIRadiusClientUsageReports

Set to **true**(1) to have the client send usage data. The default is **false**(2).

clabWIFIRadiusClientIntervalInterimReport

Set to **true**(1) to have the client periodically send Interim reports. The default is **false**(2).

clabWIFIRadiusClientAPTransitionReport

Set to **true**(1) to have the client send Transition reports when the station transitions to a different Access point. The default is **false**(2).

clabWIFIRadiusClientGigawordReport

Set to **true**(1) to have the client send Gigaword reports when the 32-bit counters rollover. The default is **false**(2).

clabWIFIRadiusClientRowStatus

The RowStatus of this entry.

MoCA MIB

The MoCA MIB provides several tables, used to configure and manage devices in the MoCA network.

mocalfConfigTable

The MoCA interface configuration table. This table supports the configuration of RF frequency, transmit power, link privacy, and traps. This table is indexed by the **ifIndex**. Touchstone firmware uses **ifIndex** 40 for the MoCA interface.

All entries are created or deleted by the firmware. The network management system cannot create or delete entries in this table.

Supported objects include:

mocalfEnable

Set to **true**(1) to enable the MoCA interface, or **false**(2) (default) to disable.

mocalfChannelMask

A bitmask, representing a list of RF center frequencies, which this MoCA node can use to form or join a MoCA network. If the new list of frequencies does not contain the frequency this MoCA node is tuned to, this node must drop from the network.

The bitmask is:

Bit	Frequency	Bit	Frequency
0	800 MHz	16	1200 MHz
1	825 MHz	17	1225 MHz
2	850 MHz	18	1250 MHz
3	875 MHz	19	1275 MHz
4	900 MHz	20	1300 MHz
5	925 MHz	21	1325 MHz

Bit	Frequency	Bit	Frequency
6	950 MHz	22	1350 MHz
7	975 MHz	23	1375 MHz
8	1000 MHz	24	1400 MHz
9	1025 MHz	25	1425 MHz
10	1050 MHz	26	1450 MHz
11	1050 MHz	27	1475 MHz
12	1100 MHz	28	1500 MHz
13	1125 MHz	29	1525 MHz
14	1150 MHz	30	1550 MHz
15	1175 MHz	31	1575 MHz

Default: **0x155540000000**.

mocalfPowerControl

Set to **true**(1) to enable automatic power control, or **false**(2) (default) to use a fixed transmit power level.

mocalfTxPowerLimit

The transmit power backoff, in dB. The transmit power changes only when the mocalfStatus object is not **linkUp**(3), which means the interface is not part of a MoCA network.

Default: **12**.

mocalfBeaconPowerLimit

The beacon transmit power backoff, in dB. The beacon transmit power changes only when the **mocalfStatus** object is not **linkUp**(3), which means the interface is not part of a MoCA network.

Default: **0**.

mocalfPowerControlTargetRate

The target transmit PHY rate, in Mbps, for the power control algorithm. Changes to this object take effect in the next maintenance cycle in the MoCA network.

Default: **235**

mocalfPrivacyEnable

Set to **true**(1) to enable link privacy and use the **mocalfPassword** object to generate the MAC management key and initial privacy management key. The default value of

false(2) disables link privacy and link encryption. Privacy cannot be enabled when the **mocalfPassword** value is empty or invalid.



Note: If this object is changed, this node drops from the network.

mocalfPassword

An ASCII numeric string, specifying the MoCA password. The string must be 12 to 17 decimal digits long. Note that while the MoCA specifications require SNMPv3 to access this object, ARRIS has extended functionality to support SNMPv1 and SNMPv2c as well.



Note: If this object is changed, this node drops from the network if **mocalfPrivacyEnable** is **true**(1).

mocalfPreferredNC

The default value of **true**(1) enables the preferred Network Controller (NC) feature on this node. To disable this feature, set it to **false**(2). This value can be ignored by the MoCA interface when operating in a MoCA 1.0 network.

mocalfStatusTable

The MoCA interface status information table. This table provides features supported, and operation parameters, of the MoCA interface. This table is indexed by the **ifIndex**.

All entries in this table are read-only.

Objects in this table include:

mocalfStatus

(read-only) The current status of the MoCA interface:

- **disabled**(1): The interface is disabled. When disabled, the MoCA interface status is unreachable except through Ethernet or WiFi (if SNMP is enabled on those interfaces).
- **noLink**(2): The interface is enabled, but not part of a network.
- **linkUp**(3): The interface is enabled and in a network.

mocalfLinkUpTime

(read-only) The time, in seconds, that this interface is part of a MoCA network. This may be used with the **sysUpTime** object to determine the link availability in the MoCA interface. Note that the value **sysUpTime** is in 10 millisecond increments, and this object is in seconds.

mocalfSoftwareVersion

(read-only) The firmware version of the MoCA device.

mocalfMocaVersion

(read-only) The MoCA version supported by this MoCA interface:

- **mocaldot0**(10)
- **mocaldot1**(11)

- **moca1dot1ProTem**(12)

mocalfNetworkVersion

(read-only) The MoCA version used in this MoCA network:

- **moca1dot0**(10)
- **moca1dot1**(11)
- **moca1dot1ProTem**(12)

If this interface is not part of a MoCA network, the value is that of mocalfMocaVersion.

mocalfMacAddress

(read-only) The MAC address of this MoCA interface. This MAC address is encoded in the first six bytes of the Globally Unique Identifier (GUID). For example, a MoCA interface with MAC address **aa: bb: cc: dd: ee: ff** has a GUID of **aa: bb: cc: dd: ee: ff: 00: 00**.

mocalfNodeID

(read-only) The node ID of this MoCA interface. If this interface is not part of a MoCA network, this object returns **0**.

mocalfName

(read-only) The textual name of this MoCA interface. In AR01.1, the interface name is "br0."

mocalfNumNodes

(read-only) The number of 1's in the GCD_BITMASK field reported in Type I Probe Reports. This value corresponds to the number of nodes that this node communicates to in the MoCA network. This value may be smaller than the number of nodes reported by the NC node. Valid range: **0** to **16**.

mocalfNC

(read-only) The node ID of the network coordinator. If this interface is not part of a MoCA network, this object returns **0**.

mocalfBackupNC

(read-only) The node ID of the backup network coordinator. If this interface is not part of a MoCA network, this object returns **0**.

mocalfRFChannel

(read-only) The MoCA channel frequency, in MHz, that this interface is tuned to when part of a MoCA network. When not part of a MoCA network, this value may not reflect the actual tuned channel. If the value of the mocalfEnable object is **false**(2), this object returns a value of **unknown**(0).

mocalfLOF

(read-only) The MoCA channel, in MHz, that this interface used when it was last in the linkUp state. If this interface has never been part of a MoCA network, this object reports the factory default Last Operational Frequency (LOF).

mocalfTabooChannelMask

(read-only) The list of taboo channels in this MoCA network, represented as a bitmask. This value is derived from TABOO_MASK_START and TABOO_CHANNEL_MASK in the

beacon, but has a different data representation. For example, if the taboo channels are 1300, 1350, and 1400 MHz:

- the TABOO_MASK_START is 52
- TABOO_CHANNEL_MASK is 'A8000000'h
- **mocalfTabooChannelMask** is '01500000'h

If this interface is not sending or receiving beacons, or there is no taboo channel in this MoCA network, this object returns a value of **0**.

mocalfNodeTabooChannelMask

(read-only) The list of taboo channels for this MoCA node, as reported in the TABOO_MASK_START and TABOO_CHANNEL_MASK fields in the node's Admission Request frame.

mocalfCapabilityMask

(read-only) The list of RF channels that this device can support, represented as a bitmask.

mocalfTxGcdPowerReduction

(read-only) The Transmit Power Control backoff, in dB, used for broadcast transmissions from this node. **mocalfTxGcdPowerReduction** is identical to the TPC backoff used for transmission, and determined from the TPC backoff parameters TPC_BACKOFF_MAJOR and TPC_BACKOFF_MINOR as follows:

mocalfTxGcdPowerReduction = TPC_BACKOFF_MAJOR * 3 + TPC_BACKOFF_MINOR

mocalfQAM256Capable

(read-only) Returns **true**(1) if this MoCA node supports QAM256, or **false**(2) if this MoCA node does not support QAM256.

mocalfPacketsAggrCapability

(read-only) The maximum number of Ethernet packets aggregated in a MoCA frame that this MoCA interface supports; one of **none**(0), **aggr6**(6), or **aggr10**(10).

mocalfMaxIngressNodeBw

(read-only) The maximum bandwidth of this MoCA interface for admission of flows, in MHz, if this node is an ingress node. This value may be obtained from REM_NODE_CAPACITY field in the Response L2ME Frame.

mocalfTxGcdRate

(read-only) The PHY rate, in Mbps, for the transmit traffic broadcast from the MoCA interface of this node.

hneMIB Objects

These objects manage and control the automatic Home Network Extender feature in Touchstone firmware.

hneWiFiGWSupport Objects

Use these objects to configure Gateway behavior for network extenders..

hneWiFiGWSearch

Set this value to **true**(1) to manually send an M-SEARCH message from the Wi-Fi Gateway.

hneWiFiGWConfigAttempts

The number of allowed configuration mismatch detections over the time period **hneWiFiGWConfigDuration** before the gateway stops attempting to configure the Home Networking Extender.

Valid Range: **1** to **255**.

Default: **10**

hneWiFiGWConfigDuration

The time, in seconds, over which the number of configuration mismatch detections per **hneWiFiGWConfigAttempts** is counted.

Default: **3600**

hneWiFiGWAutoConfigurationEnable

Set to **true**(1) to enable the ARRIS Home Networking Extender Auto-configuration support.

Default: **false**(2)

hneWiFiGWSecurityEnable

Set to **false**(2) to disable the ARRIS Home Networking Extender security support.

Default: **true**(1)

hneWiFiGWTable

This table lists Home Networking Extenders that have attempted to connect to the Wi-Fi Gateway. Each entry contains the following objects:

hneWiFiGWMACAddr

(read-only) The MAC Address of the Home Networking Extender.

hneWiFiGWIPEndrType

(read-only) The IP Address type (IPv4 or IPv6) of the Home Networking Extender.

hneWiFiGWIPEndrAddress

(read-only) The IP Address of the Home Networking Extender.

hneWiFiGWARRISAutoCfgSupport

(read-only) True if the Gateway determines the Home Networking Extender supports the ARRIS auto-configuration protocol.

hneWiFiGWLocation

(read-only) The URL from the location header field of the NOTIFY/M-SEARCH response of device discovery.

hneWiFiGWManufacturer

(read-only) The manufacturer of the Home Networking Extender from the device description during discovery.

hneWiFiGWModelName

(read-only) The model name of the Home Networking Extender from the device description during discovery.

hneWiFiGWModelNumber

(read-only) The model number of the Home Networking Extender from the device description during discovery.

hneWiFiGWConfigurationId

(read-only) Represents the current configuration of the Home Networking Extender.

Valid Range: **0** to **16777215**.

hneWiFiGWLastSynchAttemptTime

(read-only) The date and time of the last synchronization attempt of the Home Networking Extender.

hneWiFiGWLastSynchAttemptResult

(read-only) The result of the last synchronization attempt; one of:

- **u n i n i t i a l i z e d**(-1)
- **p a s s**(0)
- **f a i l H T T P S S e s s i o n E s t a b l i s h m e n t**(1)
- **f a i l H T T P S P U T**(2)

hneWiFiGWSynchedWithGW

(read-only) True if the Home Networking Extender is synchronized with the WiFi Gateway.

hneWiFiGWOverride24OutputPower

Sets the 2.4 GHz radio output power, relative to the hardware's maximum capability. This value overrides the gateway value set via `webWiFiOutputPower`. A value of 0 uses the value in the gateway configuration.

If the Home Networking Extender does not support one of the provisioned values, it should implement the closest supported value.

- **g a t e w a y D e f a u l t**(0)
- **p e r c e n t 12**(12)
- **p e r c e n t 25**(25)
- **p e r c e n t 50**(50)
- **p e r c e n t 75**(75)
- **p e r c e n t 100**(100)

Default: **g a t e w a y D e f a u l t**(0)

hneWiFiGWOverride50OutputPower

Sets the 5.0 GHz radio output power, relative to the hardware's maximum capability.

This value overrides the gateway value set via `webbWiFi50OutputPower`. A value of 0 uses the value in the gateway configuration.

If the Home Networking Extender does not support one of the provisioned values, it should implement the closest supported value.

- **gatewayDefault(0)**
- **percent12(12)**
- **percent25(25)**
- **percent50(50)**
- **percent75(75)**
- **percent100(100)**

hneWiFiGWOVERRIDE24Channel

Controls and reflects the current channel number (802.11g) or control channel (802.11n/ac). This value overrides the gateway value set via `webbWiFiChannel`.

If set to 0, the device will be put in auto-channel mode where it automatically scans for the least-crowded channel.

For 802.11g, available channels are 1-14. For 802.11n/ac, available channels are 34-216. Channel selection is also subject to restrictions based on the selected country code.

A value of 255 indicates no override and the gateway setting should be used.

Valid Range: **0** to **216**.

Default: **255**

hneWiFiGWOVERRIDE50Channel

Controls and reflects the current channel number on the 5.0 GHz radio (802.11 a/n/ac). This value overrides the gateway value set via `webbWiFi50Channel`.

Set to 0 to put the device in auto-channel mode, where it automatically scans for the least-crowded channel.

Available channels are 34 through 216, subject to restrictions based on the selected country code.

A value of 255 indicates there is no override and the gateway setting should be used.

Valid Range: **0** to **216**.

Default: **255**

hneWiFiGWOVERRIDE24ChannelBW

In 802.11n or 802.11ac mode, determines the 2.4GHz bandwidth to use:

- **gatewayDefault(-2)** (Default, use the gateway configuration value.)
- **unknown(-1)**
- **widh20MHz(0)**
- **widh40MHz(1)**
- **widh20and40Mhz(2)**

This value overrides the gateway value set via `webbWiFiChannelBW`.

hneWiFiGWOVERRIDE50ChannelBW

In 802.11n or 802.11ac mode, determines the 5.0GHz bandwidth to use:

- **gatewayDefault(-2),**
- **unknown(-1),**
- **width20MHz(0),**
- **width20and40MHz(2),**
- **width20and40and80MHz(3)**

This value overrides the gateway value set via webWiFi50ChannelBW.

hneWiFiGWSupportedTable

A table of Home Networking Extenders that are supported by the Wi-Fi GW.

hneWiFiGWSupportedManufacturer

The extender manufacturer name.

hneWiFiGWSupportedModelNumber

The extender model number.

hneWiFiGWSupportedRowStatus

Row status for the supported Home Networking Extender entry.

TR-069 Management

Touchstone firmware supports the TR-069 CPE WAN Management Protocol for provisioning and managing Touchstone Gateway products.

Overview

This section provides a brief overview of TR-069 support.



Note: For specific use of TR-069 management software, see the documentation accompanying your server.

Supported and Unsupported Features

Touchstone firmware supports the following TR-069 features:

- Provisioning of ACS connection parameters through the CM configuration file
- Supports either non-secure (HTTP) or simplified secure (HTTPS) connections
- IPv4 connections
- Router management, using the gateway IP address
- Use of DHCP to obtain ACS URL, provisioning code, CWMPRetryMinimumWaitInterval and CWMPRetryIntervalMultiplier
- Supports configuration of multiple SSIDs
- TR-181 object model

The following TR-069 features are specifically unsupported:

- IPv6 connections (IPv4 only)

- Router management through the CM IP address (gateway IP address only)
- Diagnostics, performance monitoring, or firmware download
- Full TR-069 security

Supported Servers

The Touchstone TR-069 implementation has been tested with the following TR-069 servers:

- ARRIS ECO Manage
- ARRIS EDGE
- Incognito
- OpenACS
- Alcatel Lucent (ALU) Motive
- ClearAccess

In addition, Touchstone firmware may support proprietary customer-specific servers. See the *Release Notes and Letter of Operational Considerations* for more details.

Protocol and Method Support

Touchstone firmware supports the "CWMP-1-2" protocol version, and includes the namespace string "urn:dslforum-org:cwmp-1-2" in all TR-069 messaging.

Touchstone firmware supports the following TR-069 RPC methods:

- GetRPCMethods (Mandatory)
- SetParameterValues (Mandatory)
- GetParameterValues (Mandatory)
- GetParameterNames (Mandatory)
- SetParameterAttributes (Mandatory)
- GetParameterAttributes (Mandatory)
- Reboot (Mandatory)
- FactoryReset (Optional)
- AddObject (Mandatory)
- DeleteObject (Mandatory)
- ScheduleInform (Optional)

The following TR-069 RPC Methods are not supported:

- Download (Mandatory)
- Upload (Optional)
- GetQueuedTransfers (Optional)
- GetAllQueuedTransfers (Optional)
- SetVouchers (Optional)
- GetOptions (Optional)
- ScheduleDownload (Optional)
- CancelTransfer (Optional)
- ChangeDUState (Optional)

Supported TR-181 Objects

Touchstone firmware supports the following TR-181 profiles:

- AdvancedFirewall:1
- Baseline:3
- Bridge:1
- DeviceAssociation:1
- DHCPv4Client:1
- DHCPv4Server:1
- DHCPv4ServerClientInfo:1
- DHCPv6Client:1
- DHCPv6ClientServerIdentity:1
- DHCPv6Server:1
- DHCPv6ServerClientInfo:1
- DNSRelay:1
- EthernetInterface:1
- EthernetLink:1
- GatewayInfo:1
- Hosts:2
- IPInterface:2
- IPv6Interface:1
- MoCA:1
- NAT:1
- Routing:2
- Time:1
- UPnPDev:1
- User:1
- WiFiAccessPoint:1
- WiFiRadio:1
- WiFiSSID:1
- Device.DHCPv4.Server.Pool.{i}.StaticAddress.{i}

The following sections describe the parameters available under each profile.

Baseline:3 Parameters

The Baseline:3 profile defines the parameters required for most devices.

Top-level Device. Parameters

The following Baseline parameters are available at the top-level Device. block.

RootDataModelVersion

The version of the TR-181 data model in use. Default: 2.4

InterfaceStackNumberOfEntries

The number of entries in the Device.InterfaceStack table, defined below.

Device.DeviceInfo. Parameters

These parameters provide general device information.

Manufacturer

"Arris Interactive, L.L.C."

ManufacturerOUI

An OUI (Organization Unique Identifier) for the manufacturer. For ARRIS devices, this is "0000CA"

ModelName

The model name of the device (for example, TM3402A).

Description

A basic description of the device. For Touchstone devices, the description is a variant of "ARRIS DOCSIS 3.1 / PacketCable 2.0 Touchstone Telephony Modem."

ProductClass

A string describing the product or product class.

SerialNumber

The device serial number.

HardwareVersion

The hardware revision of the device. ARRIS updates this field as needed to reflect significant hardware changes or improvements to the product.

SoftwareVersion

The firmware version of the image currently loaded on the device.

ProvisioningCode

The identifier of the primary service provider and other provisioning information. The ACS can use this information to determine service provider-specific customization and provisioning parameters.

UpTime

The time, in seconds, since the device was last restarted.

FirstUseDate

The date and UTC time when the device both first successfully established an IP-layer network connection and first acquired an absolute time reference (using NTP or equivalent) over that network connection. A factory reset clears this parameter. If the device cannot connect to an NTP server, the time is Unknown.

SupportedDataModelNumberOfEntries

The number of entries in the Device.DeviceInfo.SupportedDataModel table.

Device.DeviceInfo.MemoryStatus Parameters

These parameters describe the device physical RAM.

Total

The total physical RAM, in kb, installed on the device.

Free

The available free physical RAM, in kb.

Device.DeviceInfo.ProcessStatus Parameters

This is the head of a table describing processes running on the device.

CPUUsage

The current CPU usage, in percent.

ProcessNumberOfEntries

The number of entries in Device.DeviceInfo.ProcessStatus.Process.

Device.DeviceInfo.ProcessStatus.Process.{i} Parameters

The process table. The index is 1 to ProcessNumberOfEntries.

PID

The process ID.

Command

The name of the command starting this process.

Size

The memory, in kb, occupied by this process.

Priority

The process priority. 0 is highest.

CPUTime

The CPU time, in milliseconds, used by this process.

State

The process state; one of:

- Running
- Sleeping
- Stopped
- Idle
- Uninterruptible
- Zombie

Device.ManagementServer Parameters

These parameters describe the device association with the ACS.

EnableCWMP

Set to 1 to enable support for CPE Wan Management Protocol (CWMP).

URL

The URL of the ACS as used to connect via CWMP.

Username

The user name used to authenticate the device with the ACS.

Password

The password associated with Username.

PeriodicInformEnable

When set to 1, the device periodically sends information to the ACS, using the Inform Method call.

PeriodicInformInterval

The time, in seconds, between initiating a connection to the ACS for periodic informs.

PeriodicInformTime

The UTC time when the device begins sending periodic informs. Any specified date is ignored for the purpose of determining when to send the first periodic inform.

ParameterKey

A string, used by the ACS, to track successful changes made by the ACS. A factory reset clears this string.

ConnectionRequestURL

The URL of the device, as used by the ACS to connect to the device. For Touchstone devices, the URL is `http://gatewayIP:15627/acscal1`

ConnectionRequestUsername

The user name, used by the ACS, to authenticate a connection to this device.

ConnectionRequestPassword

The password associated with ConnectionRequestUsername.

UpgradesManaged

When set to 1, the ACS manages upgrades for the device.

DefaultActiveNotificationThrottle

The minimum wait time, in seconds, before the device initiates a connection to the ACS for active notifications. The time begins after closing the last active notification connection.

CWMPRetryMinimumWaitInterval

The minimum time, in seconds, the device waits before retrying the first connection to the ACS. Valid range: 1 to 65535.

CWMPRetryIntervalMultiplier

A multiplier used to calculate the actual wait time before retrying the connection. The actual wait time is a random number bounded by CWMPRetryMinimumWaitInterval and $(\text{CWMPRetryMinimumWaitInterval} * \text{CWMPRetryIntervalMultiplier} / 1000)$. Valid range: 1000 to 65535.

X_ARRISL_COM_ValidateManagementServerCertificate

Set to True to validate the Management Server certificate.

Device.DNS.Client Parameters

These parameters provide the head of the Device.DNS.Client.Server table.

Enable

Set to 1 (the default) to enable the DNS client.

Status

The status of this entry; one of:

- Disabled
- Enabled
- Error

ServerNumberOfEntries

The number of entries in the Device.DNS.Client.Server table.

Device.DNS Parameters

The base-level DNS parameters.

SupportedRecordTypes

The DNS record types that this device supports. One or more of the following types may be specified, separated by commas:

- A ([RFC1035])
- AAAA ([RFC3596])
- SRV ([RFC2782])
- PTR ([RFC1035])

Default: "A,AAAA"

Device.DNS.Client.Server.{i} Parameters

The DNS servers the device uses to resolve domain names. The index is 1 to ServerNumberOfEntries.

Enable

Set to True to enable this entry.

Status

The status of this entry; one of:

- Disabled
- Enabled
- Error

DNSServer

The IP address of the DNS server defined in this entry.

Interface

The name of the interface this device uses to query the DNS server.

Type

The method used to assign the DNS server address; one of:

- DHCPv4
- DHCPv6
- RouterAdvertisement
- IPCP
- Static

Device.InterfaceStack.{i} Parameters

This table describes the relationships between interfaces; in particular which interfaces run on top of other interfaces. The index is 1 to InterfaceStackNumberOfEntries.

HigherLayer

The name of an interface running on top of the interface defined in LowerLayer.

LowerLayer

The name of an interface running beneath the interface defined in HigherLayer.

Bridge:1 Parameters

The Bridge:1 profile contains Layer 2 bridging-related parameters.

Device.Bridging. Parameters

Specifies bridges between different Layer 2 interfaces.

MaxBridgeEntries

The maximum number of entries available in the Device.Bridging.Bridge table. Default: 16

MaxDBridgeEntries

The maximum number of 802.1D entries available in the Device.Bridging.Bridge table. Default: 16

BridgeNumberOfEntries

The number of entries in the Device.Bridging.Bridge table. Default: 8

Device.Bridging.Bridge.{i} Parameters

The bridge table. The index indicates a logical LAN subnet, numbered 1 through 8. (Subnets can tie together one or more physical interfaces; for example the Ethernet ports and subscriber SSIDs.)

Enable

Set to True to enable this entry.

Status

The status of this entry; one of:

- Disabled

- Enabled
- Error

PortNumberOfEntries

The number of entries in the Device.Bridging.Bridge.{i}.Port table. The default depends on the specified interface:

- Device.Bridging.Bridge.200 = 8
- Device.Bridging.Bridge.201 = 3
- Device.Bridging.Bridge.202 = 3
- Device.Bridging.Bridge.203 = 3
- Device.Bridging.Bridge.204 = 3
- Device.Bridging.Bridge.205 = 3
- Device.Bridging.Bridge.206 = 3
- Device.Bridging.Bridge.207 = 3

VLANNumberOfEntries

The number of entries in the Device.Bridging.Bridge.{i}.VLAN table.

VLANPortNumberOfEntries

The number of entries in the Device.Bridging.Bridge.{i}.VLANPort table.

Device.Bridging.Bridge.{i}.Port.{i} Parameters

The Bridge Port table for this logical LAN subnet. The index is 1 to PortNumberOfEntries.

Enable

Set to True to enable this entry.

Status

The operational state of the bridge port; one of:

- Up
- Down
- Unknown
- Dormant
- NotPresent
- LowerLayerDown
- Error

Alias

A handle, used by the ACS to refer to this entry.

Name

The name of this port as defined by the device.

LastChange

The time, in seconds, since the bridge port entered its current operational state.

LowerLayers

A comma-separated list of strings, defining the path name of an interface object stacked immediately below this interface object. The default values for each port are:

- Bridging.Bridge.200.Port.200 = "Bridging.Bridge.200.Port.6, Bridging.Bridge.200.Port.7, Bridging.Bridge.200.Port.8, Bridging.Bridge.200.Port.9, Bridging.Bridge.200.Port.40, Bridging.Bridge.200.Port.10001, Bridging.Bridge.200.Port.10101"
- Bridging.Bridge.201.Port.201 = "Bridging.Bridge.201.Port.10002, Bridging.Bridge.201.Port.10102"
- Bridging.Bridge.202.Port.202 = "Bridging.Bridge.202.Port.10003, Bridging.Bridge.202.Port.10103"
- Bridging.Bridge.203.Port.203 = "Bridging.Bridge.203.Port.10004, Bridging.Bridge.203.Port.10104"
- Bridging.Bridge.204.Port.204 = "Bridging.Bridge.204.Port.10005, Bridging.Bridge.204.Port.10105"
- Bridging.Bridge.205.Port.205 = "Bridging.Bridge.205.Port.10006, Bridging.Bridge.205.Port.10106"
- Bridging.Bridge.206.Port.206 = "Bridging.Bridge.206.Port.10007, Bridging.Bridge.206.Port.10107"
- Bridging.Bridge.207.Port.207 = "Bridging.Bridge.207.Port.10008, Bridging.Bridge.207.Port.10108"
- Bridging.Bridge.200.Port.6 = "Ethernet.Interface.6"
- Bridging.Bridge.200.Port.7 = "Ethernet.Interface.7"
- Bridging.Bridge.200.Port.8 = "Ethernet.Interface.8"
- Bridging.Bridge.200.Port.9 = "Ethernet.Interface.9"
- Bridging.Bridge.200.Port.40 = "MOCA.Interface.40"
- Bridging.Bridge.200.Port.10001 = "WiFi.SSID.10001"
- Bridging.Bridge.200.Port.10101 = "WiFi.SSID.10101"
- Bridging.Bridge.201.Port.10002 = "WiFi.SSID.10002"
- Bridging.Bridge.201.Port.10102 = "WiFi.SSID.10102"
- Bridging.Bridge.202.Port.10003 = "WiFi.SSID.10003"
- Bridging.Bridge.202.Port.10103 = "WiFi.SSID.10103"
- Bridging.Bridge.203.Port.10004 = "WiFi.SSID.10004"
- Bridging.Bridge.203.Port.10104 = "WiFi.SSID.10104"
- Bridging.Bridge.204.Port.10005 = "WiFi.SSID.10005"
- Bridging.Bridge.204.Port.10105 = "WiFi.SSID.10105"
- Bridging.Bridge.205.Port.10006 = "WiFi.SSID.10006"
- Bridging.Bridge.205.Port.10106 = "WiFi.SSID.10106"
- Bridging.Bridge.206.Port.10007 = "WiFi.SSID.10007"
- Bridging.Bridge.206.Port.10107 = "WiFi.SSID.10107"
- Bridging.Bridge.207.Port.10008 = "WiFi.SSID.10008"
- Bridging.Bridge.207.Port.10108 = "WiFi.SSID.10108"

ManagementPort

If true, this port is a management (upward-facing) bridge port. Otherwise, the port is downward-facing.

PortState

The bridge port state, as defined in 802.1D and 802.1Q; one or more of:

- Disabled
- Blocking
- Listening
- Learning
- Forwarding
- Broken

PVID

The port VLAN ID. Untagged frames arriving on this port are associated with this VLAN ID. Does not apply to 802.1D bridges.

AcceptableFrameTypes

The frame types arriving on this port that are accepted by the bridge. One or more of:

- AdmitAll
- AdmitOnlyVLANTagged
- AdmitOnlyPrioUntagged

For 802.1D operation, this parameter is always AdmitAll.

Device.Bridging.Bridge.{i}.Port.{i}.Stats Parameters

A statistics table for bridge ports. The indexes are defined as for Device.Bridging.Bridge.{i}.Port.{i} above.

BytesSent

The total number of bytes transmitted from the interface, including framing characters.

BytesReceived

The total number of bytes received on the interface, including framing characters.

PacketsSent

The total number of packets transmitted from the interface.

PacketsReceived

The total number of packets received by the interface.

ErrorsSent

The total number of packets that could not be transmitted from the interface due to errors.

ErrorsReceived

The total number of packets received by the interface that contained errors prohibiting them from being delivered to a higher-layer protocol.

UnicastPacketsSent

The total number of packets requested for transmission from the interface that were not broadcast or multicast packets. Includes packets discarded or not sent.

UnicastPacketsReceived

The total number of packets received by the interface that were not broadcast or multicast packets.

DiscardPacketsSent

The total number of outbound packets not transmitted, when there were no errors detected. This may happen, for example, to remedy a buffer overflow.

DiscardPacketsReceived

The total number of received packets that were discarded, when there were no errors detected. This may happen, for example, to remedy a buffer overflow.

MulticastPacketsSent

The total number of multicast packets transmitted from the interface. Includes packets discarded or not sent.

MulticastPacketsReceived

The total number of multicast packets received by the interface.

BroadcastPacketsSent

The total number of broadcast packets transmitted from the interface. Includes packets discarded or not sent.

BroadcastPacketsReceived

The total number of broadcast packets received by the interface.

UnknownProtoPacketsReceived

The total number of received packets discarded due to an unknown or unsupported protocol.

Device.Bridging.Bridge.{i}.VLAN.{i} Parameters

The bridge VLAN table. Applies only to 802.1Q bridges.

VLANID

The VLAN ID of the entry. Valid range: 1 to 4094.

Device.Bridging.Bridge.{i}.VLANPort.{i} Parameters

The bridge VLAN egress port and untagged port membership table. Applies only to 802.1Q bridges.

VLAN

The path name of a row in the VLAN table above.

Port

The path name of a row in the Bridge.Port table.

Untagged

When true, enables untagged port membership to the VLAN and sends egress frames untagged.

DeviceAssociation:1 Parameters

The following parameters are available under the DeviceAssociation:1 profile. In this profile, the index is a unique number assigned to each associated device.

Device.ManagementServer Parameters

This object contains parameters describing the device's association with a manageable device.

ManageableDeviceNumberOfEntries

The number of entries in the Device.ManagementServer.ManageableDevice table.

Device.ManagementServer.ManageableDevice.{i} Parameters

Entries in this table correspond to a distinct LAN device that supports device-gateway association.

ManufacturerOUI

The OUI (Organization Unique Identifier) for the LAN device.

SerialNumber

The serial number of the LAN device.

ProductClass

The product class associated with the LAN device's serial number. May be empty.

Host

A comma separated list of path names to rows in the Hosts.Host table.

DHCPv4Client:1 Parameters

The following parameters are available under the DHCPv4Client:1 profile.

Device.DHCPv4 Parameters

ClientNumberOfEntries

The number of entries in the Device.DHCPv4.Client table.

Device.DHCPv4.Client.{i} Parameters

This table provides DHCPv4 client settings for the specified interface.

Enable

Set to True to enable this entry..

Interface

The path name to an interface in the Device.IP.Interface table.

Status

The status of this entry; one of:

- Disabled
- Enabled
- Error_Misconfigured
- Error

Renew

When set to true, the DHCP client renews its lease.

IPAddress

The IPv4 address option received from the DHCP server.

SubnetMask

The subnet mask option received from the DHCP server.

IPRouters

A comma-separated list of IP router IPv4 addresses, received from DHCP Options 3, 33, or 121.

DNSServers

A comma-separated list of DNS server IPv4 addresses, received from DHCP Option 6.

LeaseTimeRemaining

The remaining DHCP lease time, in seconds. If set to -1, the lease is indefinite.

SentOptionNumberOfEntries

The number of entries in the Device.DHCPv4.Client.SentOption table.

ReqOptionNumberOfEntries

The number of entries in the Device.DHCPv4.Client.ReqOption table.

Device.DHCPv4.Client.{i}.SentOption.{i} Parameters

This table represents DHCPv4 options that must, if enabled, be sent in DHCPv4 client requests.

Enable

Set to True to enable this entry.

Tag

An option tag as defined in RFC 2132. Valid range: 1 to 254.

Value

A hexbinary encoded value associated with this option.

Device.DHCPv4.Client.{i}.ReqOption.{i} Parameters

This table represents DHCPv4 options that must, if enabled, be requested in DHCPv4 client requests.

Enable

Set to True to enable this entry.

Order

The position of this option in the DHCP request. A value of 1 indicates the first entry.

Tag

An option tag as defined in RFC 2132. Valid range: 1 to 254.

Value

A hexbinary encoded value associated with this option.

DHCPv4Server:1 Parameters

The following parameters are available under the DHCPv4Server:1 profile.

Device.DHCPv4.Server Parameters

This is the head of the DHCPv4 server configuration.

Enable

Set to true to enable the DHCP server.

PoolNumberOfEntries

The number of entries in the Device.DHCPv4.Server.Pool table. The default configuration provides 8 entries.

Device.DHCPv4.Server.Pool.{i} Parameters

Each entry in this table describes a conditional DHCPv4 server pool.

Enable

Set to True to enable this entry.

Order

Position of this pool entry, sorted by precedence. A value of 1 indicates the highest precedence.

Interface

A path name to a row in the Device.IP.Interface table.

MinAddress

Specifies the first IPv4 address in the pool to be assigned by the DHCP server on the LAN interface.

MaxAddress

Specifies the last IPv4 address in the pool to be assigned by the DHCP server on the LAN interface.

ReservedAddresses

A comma-separated list (up to 32 items) of IPv4Addresses. Each address in this list is marked reserved from the address allocation pool.

SubnetMask

Specifies the client's network subnet mask.

DNSServers

A comma-separated list (up to 4 items) of IPv4Addresses. List items represent DNS servers offered to DHCP clients.

DomainName

The domain name to provide to clients on the LAN interface.

IPRouters

A comma-separated list (up to 4 items) of IPv4Addresses. Each address is that of a router (default gateway) on this subnet.

LeaseTime

The lease time, in seconds, of client assigned addresses. A value of -1 indicates an infinite lease.

OptionNumberOfEntries

The number of entries in the Device.DHCPv4.Server.Pool.Option table.

ClientNumberOfEntries

The number of entries in the Device.DHCPv4.Server.Pool.Client table.

Device.DHCPv4.Server.Pool.{i}.Option.{i}.

Each entry in this table specifies an option that is returned to clients whose DHCP requests are associated with this pool.

Enable

Set to True to enable this entry.

Tag

An option tag as defined in RFC 2132. Valid range: 1 to 254.

Value

A hexbinary encoded value associated with this option.

DHCPv4ServerClientInfo:1 Parameters

The following parameters are available under the DHCPv4ServerClientInfo:1 profile.

Device.DHCPv4.Server.Pool{i}.Client.{i} Parameters**Chaddr**

The MAC address of the DHCPv4 client.

Active

True if the DHCPv4 client is currently present on the LAN.

IPv4AddressNumberOfEntries

The number of entries in the Device.DHCPv4.Server.Pool.{i}.Client.{i}.IPv4Address table.

OptionNumberOfEntries

The number of entries in the Device.DHCPv4.Server.Pool.{i}.Client.{i}.Option table.

Device.DHCPv4.Server.Pool.{i}.Client.{i}.IPv4Address.{i} Parameters

Each entry in this table provides information about an IPv4 address assigned to the client.

IPAddress

The IPv4 address assigned to the client.

LeaseTimeRemaining

The time when the DHCP lease expires, or 0001-01-01T00:00:00Z if not known. For an infinite lease, the parameter value is 9999-12-31T23:59:59Z.

Device.DHCPv4.Server.Pool{i}.Client.{i}.Option.{i} Parameters

Each entry in this table contains a DHCPv4 option supplied by the client.

Tag

An option tag as defined in RFC 2132. Valid range: 1 to 254.

Value

A hexbinary encoded value associated with this option.

Device.DHCPv4.Server.Pool.{i}.StaticAddress.{i} Parameters

The DHCP static address table. Entries in this table describe client addresses assigned by the network administrator, and DHCP is used to pass the assigned address to the client.

Enable

Set to True to enable this entry.

Alias

A handle, used by the ACS to refer to this entry.

Chaddr

The MAC address of the client physical interface.

Yiaddr

The static IPv4 address assigned to the client.

DHCPv6Client:1 Parameters

The following parameters are available under the DHCPv6Client:1 profile.

Device.DHCPv6. Parameters

The DHCPv6 object. It provides client and server entries.

ClientNumberOfEntries

The number of entries in the Device.DHCPv6.Client. table.

Device.DHCPv6.Client.{i}. Parameters

Each entry in this table contains DHCPv6 client settings for an associated IP interface.

Enable

Set to True to enable this entry.

Status

The status of this entry; one of:

- Disabled
- Enabled
- Error_Misconfigured
- Error

Interface

The path name to a row in the IP.Interface table.

DUID

The client's DHCP Unique Identifier (DUID).

RequestAddresses

Set to true (the default) to enable inclusion of the Identity Association (IA) for the Non-temporary Address option in Solicit messages.

RequestPrefixes

Set to true (the default) to enable inclusion of the (IA) for Prefix Delegation option in Solicit messages.

This is only appropriate for an upstream interface on a requesting router, e.g. for an RG WAN interface.

RapidCommit

Set to true (the default) to enable inclusion of the Rapid Commit option in Solicit messages.

Renew

When set to true, the client renews its DHCPv6-supplied information.

SuggestedT1

The T1 timer. The value, in seconds, the client should use when sending IA options. Set to -1 (the default) to indicate no T1 value is specified.

SuggestedT2

The T2 timer. The value, in seconds, the client should use when sending IA options. Set to -1 (the default) to indicate no T2 value is specified.

SupportedOptions

A comma-separated list of client-supported options, including both top-level and encapsulated options.

RequestedOptions

A comma-separated list of top-level options that the client explicitly requests from the server.

ServerNumberOfEntries

The number of entries in the Device.DHCPv6.Client.{i}.Server table.

DHCPv6ClientServerIdentity:1 Parameters

The following parameters are available under the DHCPv6ClientServerIdentity:1 profile.

Device.DHCPv6.Client{i}.Server.{i} Parameters

Each entry in this table lists a discovered DHCPv6 server.

SourceAddress

The IPv6 address from which the last message received from this server was sent.

DUID

The server's DHCP Unique Identifier (DUID), as received in the ServerID option.

InformationRefreshTime

The value of the last Information Refresh Time option received from this server, converted to the date and time when the associated information expires.

DHCPv6Server:1 Parameters

The following parameters configure the DHCPv6 server.

Device.DHCPv6.Server Parameters**Enable**

Set to true to enable the DHCPv6 server.

PoolNumberOfEntries

The number of entries in the Device.DHCPv6.Server.Pool table.

Device.DHCPv6.Server.Pool.{i} Parameters

The DHCPv6 server pool table. The Pool index uses the ifIndex numbers, 200 through 207.

Enable

Set to True to enable this entry.

Status

The status of this entry; one of:

- Disabled
- Enabled
- Error_Misconfigured
- Error

Interface

A path name to a row in the Device.IP.Interface table.

IANAEnable

Set to true to enable IANA offers.

IANAPrefixes

A comma-separated list of path names to rows in the Device.IP.Interface.{i}.IPv6Prefix table, listing prefixes from which IA_NA addresses are assigned.

DUID

(unused)

VendorClassID

(unused)

UserClassID

(unused)

SourceAddress

(unused)

SourceAddressMask

(unused)

Order

The position of the pool entry, in order of precedence. The lowest number is the highest preference. By default, the order is the same as the interface index of each entry in the pool (200-207).

IAPDEnable

Set to true to enable IAPD offers. Disabled by default.

IAPDAddLength

The recommended minimum number of bits to add to IAPD prefixes to determine the length of prefixes offered in an IA-PD message. The default is 0.

ClientNumberOfEntries

The number of rows in the Device.DHCPv6.Server.Pool.Client table.

OptionNumberOfEntries

The number of rows in the Device.DHCPv6.Server.Pool.Option table.

Device.DHCPv6.Server.Pool.{i}.Option.{i} Parameters

Entries in this table specify DHCPv6 options that must (if enabled) be offered to clients whose requests are associated with this pool.

Enable

Set to True to enable this entry.

Tag

The option code.

Value

A hexbinary encoded option value.

DHCPv6ServerClientInfo:1 Parameters

The following parameters are available under the DHCPv6ServerClientInfo:1 profile.

Device.DHCPv6.Server.Pool.{i}.Client.{i} Parameters

These parameters point to address and option tables.

SourceAddress

The source address of the DHCPv6 client.

Active

True if the DHCPv6 client is currently present on the LAN.

IPv6AddressNumberOfEntries

The number of entries in the Device.DHCPv6.Server.Pool.{i}.Client.{i}.IPv6Address table.

IPv6PrefixNumberOfEntries

The number of entries in the Device.DHCPv6.Server.Pool.{i}.Client.{i}.IPv6Prefix table.

OptionNumberOfEntries

The number of entries in the Device.DHCPv6.Server.Pool.{i}.Client.{i}.Option table.

Device.DHCPv6.Server.Pool.{i}.Client.{i}.IPv6Address.{i} Parameters

This table provides IPv6 addresses assigned to the client.

IPAddress

The assigned IPv6 address.

PreferredLifetime

(unused)

ValidLifetime

(unused)

Device.DHCPv6.Server.Pool.{i}.Client.{i}.Option{i} Parameters

This table contains DHCPv6 options supplied by this client.

Tag

The DHCP Option code.

Value

A hexbinary encoded option value.

DNSRelay:1 Parameters

The following parameters are available under the DNSRelay:1 profile.

Device.DNS.Relay Parameters

Configures and manages the DNS Relay function, which forwards local network DNS queries to local or external DNS servers.

Enable

Set to True to enable DNS Relay.

Status

The status of this entry; one of:

- Disabled
- Enabled
- Error

ForwardNumberOfEntries

The number of entries in the Device.DNS.Relay.Forwarding table.

Device.DNS.Relay.Forwarding.{i} Parameters

Rows in this table define a DNS Server forwarding policy for the DNS Relay.

Enable

Set to True to enable this entry.

Status

The status of this entry; one of:

- Disabled
- Enabled
- Error

DNSServer

The DNS server address to receive forwarded queries.

Interface

A path name to a row in the Device.IP.Interface table, specifying the interface used to forward DNS queries. Set to an empty string to use the device routing policy.

Type

The method used to assign the DNS server address; one of:

- DHCP (Replaced by DHCPv4, DEPRECATED)
- DHCPv4
- DHCPv6
- RouterAdvertisement
- IPCP
- Static

Download:1 Parameters

The following parameters are available under the Download:1 profile. They support the download phase of a TR-143 Speedtest.

Device.IP.Diagnostics.DownloadDiagnostics Parameters

These parameters define the configuration for a HTTP and FTP Download Speedtest.

DiagnosticsState

Indicates the availability of diagnostic data; one of:

- None
- Requested
- Completed
- Error_InitConnectionFailed
- Error_NoResponse
- Error_TransferFailed
- Error_PasswordRequestFailed
- Error_LoginFailed
- Error_NoTransferMode
- Error_NoPASV
- Error_IncorrectSize
- Error_Timeout

Only Requested may be written, to start the download test. When the test completes, the value of this parameter is either Completed (if the test completed successfully), or one of the Error values listed above. If Requested is written while a test is in progress, the test restarts.

To stop a test in progress, write to any of the other Download parameters. In this situation, the DiagnosticsState is None.

Interface

The path name to a row in the Device.IP.Interface table, specifying the interface used for the download test. Use an empty string to specify the default routing interface.

DownloadURL

An HTTP or FTP URL, identifying a resource to download. The following considerations apply:

- An FTP download uses binary mode.
- An HTTP download uses persistent connections. Pipelining and HTTP Authentication are not used.

DownloadTransports

A comma-separated list of strings, identifying supported download transport protocols for a CPE device. Typically, the string is "HTTP,FTP"

DSCP

The DiffServ code point for marking packets transmitted in the test.

EthernetPriority

The Ethernet priority code for marking packets transmitted in the test (if applicable).

ROMTime

The request time, in UTC, specified to microsecond precision. For example: 2008-04-09T15:01:05.123456

For HTTP, this is the time when the device sent the GET command.

For FTP, this is the time when the device sent the RTRV command.

BOMTime

The time, in UTC, when transmission began, specified to microsecond precision. For example: 2008-04-09T15:01:05.123456

For HTTP, this is the time when the first data packet was received.

For FTP, this is the time when the client received the first data packet on the data connection.

EOMTime

The time, in UTC, when the transmission ended, specified to microsecond precision. For example: 2008-04-09T15:01:05.123456

For HTTP, this is the time when the last data packet was received.

For FTP, this is the time when the client received the last packet on the data connection.

TestBytesReceived

The test traffic received, in bytes, during the FTP/HTTP transaction. Includes FTP/HTTP headers. The test is bounded by BOMTime and EOMTime.

TotalBytesReceived

The total number of bytes received on the Interface between BOMTime and EOMTime.

TCPOpenRequestTime

The request time, in UTC, specified to microsecond precision. For example: 2008-04-09T15:01:05.123456

For HTTP, this is the time when the TCP socket open (SYN) was sent for the HTTP connection.

For FTP, this is the time when the TCP socket open (SYN) was sent for the data connection.

TCPOpenResponseTime

The response time, in UTC, specified to microsecond precision.

For HTTP, this is the time when the TCP ACK to the socket opening the HTTP connection was received.

For FTP, this is the time when the TCP ACK to the socket opening the data connection was received.

DSLite:1 Parameters

These parameters configure and route IPv6 Dual-Stack Lite (DS-Lite). DS-Lite can be used when the Gateway has only IPv6 connectivity to the WAN, but needs to support LAN devices that support only IPv4 connectivity.

Device.DSLite Parameters

The top-level DS-Lite parameters.

InterfaceSettingNumberOfEntries

The number of entries in the InterfaceSetting table.

Device.DSLite.InterfaceSetting.{i} Parameters

DS-Lite interface settings.

Enable

Set to True to enable this entry..

Status

The status of this entry; one of:

- Disabled
- Enabled
- Error

EndpointAssignmentPrecedence

The preferred method to use when assigning values to EndpointName and EndpointAddress, when both static and dynamic values are available to them. One of:

- DHCPv6
- Static

EndpointName

The Fully Qualified Domain Name (FQDN) of the tunnel concentrator (remote endpoint).

EndpointAddress

The IPv6 address of the tunnel concentrator (remote endpoint).

Origin

Method used to assign EndpointName and EndpointAddress; one of:

- DHCPv6 (assigned by DHCPv6)
- Static (For example, present in the factory default configuration, set by the ACS, or set by some other management entity such as a web-based configuration interface)

EthernetInterface:1 Parameters

The following parameters are available under the EthernetInterface:1 profile.

Device.Ethernet. Parameters

The Ethernet interface configuration on this device.

InterfaceNumberOfEntries

The number of entries in the Device.Ethernet.Interface table. Touchstone Gateway devices typically have four Ethernet ports, each of which has an entry.

Device.Ethernet.Interface.{i} Parameters

This table provides configuration data for each Ethernet port.

Enable

Set to true to enable this Ethernet port.

Status

Ethernet port status; one of:

- Up
- Down
- Unknown
- Dormant
- NotPresent
- LowerLayerDown
- Error

Alias

A handle, used by the ACS to refer to this entry.

Name

The name of the Ethernet port, assigned by the device.

LastChange

The time, in seconds, since the port entered its current operating state.

Upstream

Always False since the Ethernet ports are LAN interfaces.

MACAddress

The MAC address of the port.

MaxBitRate

The maximum upstream and downstream PHY bit rate for this port, in Mbps.

DuplexMode

The duplex mode for the connection; one of:

- Half
- Full
- Auto

Device.Ethernet.Interface.{i}.Stats Parameters

Throughput statistics for this Ethernet port.

BytesSent

The total number of bytes transmitted from the interface, including framing characters.

BytesReceived

The total number of bytes received on the interface, including framing characters.

PacketsSent

The total number of packets transmitted from the interface.

PacketsReceived

The total number of packets received on the interface.

ErrorsSent

The total number of outbound packets that could not be transmitted because of errors.

ErrorsReceived

The total number of inbound packets that contained errors, preventing them from being delivered to a higher-layer protocol.

UnicastPacketsSent

The total number of packets requested for transmission which were not addressed to a multicast or broadcast address at this layer, including those that were discarded or not sent.

UnicastPacketsReceived

The total number of received packets, delivered by this layer to a higher layer, which were not addressed to a multicast or broadcast address at this layer.

DiscardPacketsSent

The total number of outbound packets which were discarded even though no errors had been detected to prevent their being transmitted. (For example, to free up buffer space.)

DiscardPacketsReceived

The total number of inbound packets which were discarded even though no errors had been detected to prevent their being delivered. (For example, to free up buffer space.)

MulticastPacketsSent

The total number of packets that higher-level protocols requested for transmission and which were addressed to a multicast address at this layer, including those that were discarded or not sent.

MulticastPacketsReceived

The total number of received packets, delivered by this layer to a higher layer, which were addressed to a multicast address at this layer.

BroadcastPacketsSent

The total number of packets that higher-level protocols requested for transmission and which were addressed to a broadcast address at this layer, including those that were discarded or not sent.

BroadcastPacketsReceived

The total number of received packets, delivered by this layer to a higher layer, which were addressed to a broadcast address at this layer.

UnknownProtoPacketsReceived

The total number of packets received via the interface which were discarded because of an unknown or unsupported protocol.

EthernetLink:1 Parameters

The following parameters are available under the EthernetLink:1 profile. In this profile, the index is 3 for the eRouter WAN interface, or 1, 2, or 4-8 for logical LAN interfaces.

Device.Ethernet. Parameters

The Ethernet interface configuration on this device.

LinkNumberOfEntries

The number of entries in the Device.Ethernet.Link table.

Device.Ethernet.Link.{i} Parameters

The Ethernet link layer table. Entries in this table model the Logical Link Control (LLC) layer of the Ethernet interface.

Enable

Set to True to enable this entry.

Status

Link status; one of:

- Up
- Down
- Unknown
- Dormant
- NotPresent
- LowerLayerDown
- Error

Alias

A handle, used by the ACS to refer to this entry.

Name

The link name as assigned by the device.

LastChange

The time, in seconds, since the link entered its current state.

LowerLayers

A comma-separated list of path names to interface objects stacked immediately below this link.

MACAddress

The MAC address associated with this link.

Device.Ethernet.Link.{i}.Stats Parameters

Link-level throughput statistics.

BytesSent

The total number of bytes transmitted from the interface, including framing characters.

BytesReceived

The total number of bytes received on the interface, including framing characters.

PacketsSent

The total number of packets transmitted from the interface.

PacketsReceived

The total number of packets received on the interface.

ErrorsSent

The total number of outbound packets that could not be transmitted because of errors.

ErrorsReceived

The total number of inbound packets that contained errors, preventing them from being delivered to a higher-layer protocol.

UnicastPacketsSent

The total number of packets requested for transmission which were not addressed to a multicast or broadcast address at this layer, including those that were discarded or not sent.

UnicastPacketsReceived

The total number of received packets, delivered by this layer to a higher layer, which were not addressed to a multicast or broadcast address at this layer.

DiscardPacketsSent

The total number of outbound packets which were discarded even though no errors had been detected to prevent their being transmitted. (For example, to free up buffer space.)

DiscardPacketsReceived

The total number of inbound packets which were discarded even though no errors had been detected to prevent their being delivered. (For example, to free up buffer space.)

MulticastPacketsSent

The total number of packets that higher-level protocols requested for transmission and which were addressed to a multicast address at this layer, including those that were discarded or not sent.

MulticastPacketsReceived

The total number of received packets, delivered by this layer to a higher layer, which were addressed to a multicast address at this layer.

BroadcastPacketsSent

The total number of packets that higher-level protocols requested for transmission and which were addressed to a broadcast address at this layer, including those that were discarded or not sent.

BroadcastPacketsReceived

The total number of received packets, delivered by this layer to a higher layer, which were addressed to a broadcast address at this layer.

UnknownProtoPacketsReceived

The total number of packets received via the interface which were discarded because of an unknown or unsupported protocol.

GatewayInfo:1 Parameters

The following parameters are available under the GatewayInfo:1 profile.

Device.GatewayInfo Parameters

Provides information associated with an Internet gateway device.

ManufacturerOUI

The Organizationally Unique Identifier (OUI) used to identify the device manufacturer.

ProductClass

Product class identifier of the gateway device.

SerialNumber

The serial number of the gateway device.

Hosts:2 Parameters

The following parameters are available under the Hosts:2 profile.

Device.Hosts Parameters

These parameters provide information about each host on the LAN.

HostNumberOfEntries

Number of entries in the Device.Hosts.Host table.

Device.Hosts.Host.{i} Parameters

Entries in this table describe each LAN device.

IPAddress

The current IP address of the host.

LeaseTimeRemaining

The time, in seconds, remaining for the current DHCP lease.

PhysAddress

The MAC address of the host.

DHCPClient

A comma-separated list, up to two strings. Each item is a path name to an entry in the

Device.DHCPv4.Server.Pool.{i}.Client or Device.DHCPv4.Server.Pool.{i}.Client table, indicating the DHCP server's client entry associated with this host.

HostName

The device's host name. May be an empty string.

Active

True if the host is currently present on the LAN.

Layer1Interface

A path name to a row in a layer 1 interface table. Example: Device.Ethernet.Interface.2

Layer3Interface

A path name to a row in the Device.IP.Interface table.

ClientID

A hexbinary string, containing the value of the Client Identifier DHCP option (Option 61) for the IP connection.

AssociatedDevice

A path name to an AssociatedDevice (or equivalent) table that models the host. Example: Device.WiFi.AccessPoint.1.AssociatedClient.2

IPv4AddressNumberOfEntries

The number of entries in the Device.Hosts.Host.{i}.IPv4Address table.

IPv6AddressNumberOfEntries

The number of entries in the Device.Hosts.Host.{i}.IPv6Address table.

UserClassID

A hexbinary string, containing the value of the host's User Class Identifier DHCP option (Option 77).

VendorClassID

A hexbinary string, containing the value of the host's Vendor Class Identifier DHCP option (Option 60).

Device.Hosts.Host.{i}.IPv4Address.{i} Parameter

This table provides known IPv4 addresses for the host.

IPAddress

The host IPv4 address.

Device.Hosts.Host.{i}.IPv6Address.{i} Parameter

This table provides known IPv4 addresses for the host.

IPAddress

The host IPv6 address.

IPInterface:2 Parameters

The following parameters are available under the IPInterface:2 profile.

Device.IP Parameters

Provides Interface, ActivePort, and Diagnostics information.

IPv4Enable

True if the IPv4 stack is enabled.

IPv4Status

The IPv4 stack status; one of:

- Disabled
- Enabled
- Error

IPv4Capable

True if the device is IPv4-capable.

IPv6Capable

True if the device is IPv6-capable.

IPv6Enable

True if the IPv6 stack is enabled on this device.

IPv6Status

The IPv6 stack status; one of:

- Disabled
- Enabled
- Error

ULAPrefix

For IPv6-capable hosts, the ULA /48 prefix.

InterfaceNumberOfEntries

The number of entries in the Device.IP.Interface table.

ActivePortNumberOfEntries

The number of entries in the Device.IP.ActivePort table.

Device.IP.Interface.{i} Parameters

Layer 3 IP Interface table.

Enable

Set to True to enable this entry.

IPv4Enable

Set to true to attach this interface to the IPv4 stack.

IPv6Enable

Set to true to attach this interface to the IPv6 stack.

Status

The current operational state of the interface; one of:

- Up
- Down
- Unknown
- Dormant
- NotPresent
- LowerLayerDown
- Error

Alias

A handle, used by the ACS to refer to this entry.

Name

The interface name assigned by the device itself.

LastChange

The time, in seconds, since the interface entered its current operational state.

LowerLayers

A comma-separated list of path names of interface objects stacked directly below the IP interface.

Type

The IP interface type; one of:

- Normal (default)
- Loopback
- Tunnel
- Tunneled

ULAEnable

True if ULAs are generated and used on this interface.

Reset

Set to True to tear down the existing IP connection and establish a new one.

IPv4AddressNumberOfEntries

Number of entries in the Device.IP.Interface.{i}.IPv4Address.{i} table.

IPv6AddressNumberOfEntries

Number of entries in the Device.IP.Interface.{i}.IPv6Address.{i} table.

IPv6PrefixNumberOfEntries

Number of entries in the Device.IP.Interface.{i}.IPv6Prefix.{i} table.

AutoIPEnable

If true, enable auto-IP on the interface (IPv4 only).

Loopback

If true, the IP interface is a loopback interface. Setting this parameter to True sets the Type to Loopback and clears the LowerLayers parameter.

MaxMTUSize

The maximum transmission unit (MTU), the largest allowed size of the IP packet transmitted by this device. Valid range: 64 to 65535.

Router

Path name to a row in the Device.Routing.Router table, indicating the routing address for this device.

Device.IP.Interface.{i}.IPv4Address.{i} Parameters

The IPv4 address table for this device. Represents IPv4 addresses assigned by DHCP, auto-IP, or statically.

Status

The status of this entry; one of:

- Disabled
- Enabled
- Error_Misconfigured
- Error

IPAddress

The IPv4 address for this device.

SubnetMask

The IPv4 subnet mask for this device.

AddressingType

The method used to assign the IP address; one of:

- DHCP
- AutoIP
- IPCP
- Static

Device.IP.Interface.{i}.Stats. Parameters

Throughput statistics for this interface.

BytesSent

The total number of bytes transmitted from the interface, including framing characters.

BytesReceived

The total number of bytes received on the interface, including framing characters.

PacketsSent

The total number of packets transmitted from the interface.

PacketsReceived

The total number of packets received on the interface.

ErrorsSent

The total number of outbound packets that could not be transmitted because of errors.

ErrorsReceived

The total number of inbound packets that contained errors, preventing them from being delivered to a higher-layer protocol.

UnicastPacketsSent

The total number of packets requested for transmission which were not addressed to a multicast or broadcast address at this layer, including those that were discarded or not sent.

UnicastPacketsReceived

The total number of received packets, delivered by this layer to a higher layer, which were not addressed to a multicast or broadcast address at this layer.

DiscardPacketsSent

The total number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. (For example, to free up buffer space.)

DiscardPacketsReceived

The total number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being delivered. (For example, to free up buffer space.)

MulticastPacketsSent

The total number of packets that higher-level protocols requested for transmission and which were addressed to a multicast address at this layer, including those that were discarded or not sent.

MulticastPacketsReceived

The total number of received packets, delivered by this layer to a higher layer, which were addressed to a multicast address at this layer.

BroadcastPacketsSent

The total number of packets that higher-level protocols requested for transmission and which were addressed to a broadcast address at this layer, including those that were discarded or not sent.

Note that IPv6 does not define broadcast addresses, so IPv6 packets never cause this counter to increment.

BroadcastPacketsReceived

The total number of received packets, delivered by this layer to a higher layer, which were addressed to a broadcast address at this layer.

Note that IPv6 does not define broadcast addresses, so IPv6 packets never cause this counter to increment.

UnknownProtoPacketsReceived

The total number of packets received via the interface which were discarded because of an unknown or unsupported protocol.

IPv6Interface:1 Parameters

The following parameters are available under the IPv6Interface:1 profile.

Device.IP Parameters

The IPv6-related parameters.

IPv6Capable

True if the device is IPv6-capable.

IPv6Enable

Set to True to enable the IPv6 stack.

IPv6Status

The IPv6 stack status; one of:

- Disabled
- Enabled
- Error

ULAPrefix

The ULA /48 prefix.

InterfaceNumberOfEntries

The number of entries in the Device.IP.Interface table.

Device.IP.Interface.{i} Parameters

The IPv6-related parameters in the IP interface table.

Enable

Set to True to enable the interface.

IPv6Enable

Set to True to attach the IPv6 stack to this interface.

Status

The current operating state of this interface; one of:

- Up
- Down
- Unknown
- Dormant
- NotPresent
- LowerLayerDown
- Error

Alias

A handle, used by the ACS to refer to this entry.

Name

The name of this interface, as assigned by the device.

LastChange

The time, in seconds, since the interface entered its current operational state.

LLowerLayers

A comma-separated list of path names of interface objects stacked directly below the IP interface.

Type

The IP interface type; one of:

- Normal (default)
- Loopback
- Tunnel
- Tunneled

ULAEnable

Set to True to generate and use ULAs (as defined in RFC 4193) on this interface.

Reset

Set to True to tear down the existing IP connection and establish a new one.

IPv4AddressNumberOfEntries

Number of entries in the Device.IP.Interface.{i}.IPv4Address.{i} table.

IPv6AddressNumberOfEntries

Number of entries in the Device.IP.Interface.{i}.IPv6Address.{i} table.

Device.IP.Interface.{i}.IPv6Address.{i} Parameters

The IPv6 address table for this device. Represents IPv6 addresses assigned to the device.

Enable

Set to True to enable this entry.

Status

The status of this entry; one of:

- Disabled
- Enabled
- Error_Misconfigured
- Error

IPAddressStatus

The address status; one of:

- Preferred (Valid address that can appear as the destination or source address of a packet)
- Deprecated (Valid but deprecated address that is not intended to be used as a source address)
- Invalid (Invalid address that is not intended to appear as the destination or source address of a packet)

- Inaccessible (Valid address that is not accessible because the interface to which it is assigned is not operational)
- Unknown (Address status cannot be determined for some reason)
- Tentative (The uniqueness of the address on the link is being verified)
- Duplicate (Invalid address that has been determined to be non-unique on the link)
- Optimistic (Valid address that is available for use, subject to restrictions, while its uniqueness on a link is being verified)

IPAddress

The IPv6 address.

Origin

Indicates how the IP address was assigned; one of:

- AutoConfigured (default) Automatically generated. For example, a link-local address as specified by SLAAC, a global address as specified by SLAAC, or generated by the CPE (often from a delegated prefix or a ULA /48 prefix).
- DHCPv6 (Assigned by DHCPv6)
- WellKnown (Specified by a standards organization; for example, the ::1 loopback address)
- Static (Static address. Examples include: present in the factory default configuration, but not WellKnown, created by the ACS, or created by some other management entity such as the eRouter's web-based interface)

Prefix

Path name to a row in the Device.IP.Interface.{i}.IPv6Prefix table.

PreferredLifetime

The date and time at which this address becomes deprecated (no longer preferred).

ValidLifetime

The date and time at which this address becomes invalid.

Anycast

True if this is an anycast address.

Device.IP.Interface.{i}.IPv6Prefix Parameters

The interface's IPv6 prefixes.

Enable

Set to True to enable this entry.

Status

The status of this entry; one of:

- Disabled
- Enabled
- Error

PrefixStatus

The prefix status, indicating whether it can be used; one of:

- Preferred (Valid prefix)
- Deprecated (Valid but deprecated prefix)
- Invalid (Invalid prefix)
- Inaccessible (Valid prefix that is not accessible because the interface to which it is assigned is not operational)
- Unknown (Prefix status cannot be determined)

Prefix

The IPv6 address prefix.

Origin

Indicates how the prefix was assigned; one of:

- AutoConfigured (default) Automatically generated. For example, generated by the CPE (using the ULA /48 prefix) or from an internal prefix not listed in this table.
- PrefixDelegation (upstream interfaces only) Delegated by DHCPv6 or some other protocol such as IPv6rd.
- RouterAdvertisement (upstream interfaces only) Discovered by router advertisement Prefix Information Option.
- WellKnown (Specified by a standards organization; for example, fe80::/10 for link-local addresses, or ::1/128 for the loopback address)
- Static (downstream interfaces only) Static address. Examples include: present in the factory default configuration, but not WellKnown, created by the ACS, or created by some other management entity such as the eRouter's web-based interface.
- Child (downstream interfaces only) Derived from an associated AutoConfigured or PrefixDelegation parent prefix.

OnLink

True if this prefix can be used for on-link determination.

Autonomous

True if this prefix can be used for generating global addresses as specified by SLAAC.

PreferredLifetime

The date and time this prefix is no longer preferred and becomes deprecated.

ValidLifetime

The date and time this prefix is no longer valid and becomes invalid.

StaticType

For static prefixes, the prefix sub-type; one of:

- Static (a normal Static prefix)
- Inapplicable (prefix is not Static)
- PrefixDelegation (prefix is populated when a PrefixDelegation prefix needs to be created)
- Child (prefix is populated when a Child prefix needs to be created)

ChildPrefixBits

A prefix that specifies the length of static Child prefixes, and how they are derived from the Parent prefix.

ParentPrefix

Path name to another row in this table, specifying the parent prefix from which this prefix was derived. Applies only to Child and static Child prefixes.

Device.IP.Interface.{i}.Stats. Parameters

Throughput statistics for this interface.

BytesSent

The total number of bytes transmitted from the interface, including framing characters.

BytesReceived

The total number of bytes received on the interface, including framing characters.

PacketsSent

The total number of packets transmitted from the interface.

PacketsReceived

The total number of packets received on the interface.

ErrorsSent

The total number of outbound packets that could not be transmitted because of errors.

ErrorsReceived

The total number of inbound packets that contained errors, preventing them from being delivered to a higher-layer protocol.

UnicastPacketsSent

The total number of packets requested for transmission which were not addressed to a multicast or broadcast address at this layer, including those that were discarded or not sent.

UnicastPacketsReceived

The total number of received packets, delivered by this layer to a higher layer, which were not addressed to a multicast or broadcast address at this layer.

DiscardPacketsSent

The total number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. (For example, to free up buffer space.)

DiscardPacketsReceived

The total number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being delivered. (For example, to free up buffer space.)

MulticastPacketsSent

The total number of packets that higher-level protocols requested for transmission and

which were addressed to a multicast address at this layer, including those that were discarded or not sent.

MulticastPacketsReceived

The total number of received packets, delivered by this layer to a higher layer, which were addressed to a multicast address at this layer.

BroadcastPacketsSent

The total number of packets that higher-level protocols requested for transmission and which were addressed to a broadcast address at this layer, including those that were discarded or not sent.

Note that IPv6 does not define broadcast addresses, so IPv6 packets never cause this counter to increment.

BroadcastPacketsReceived

The total number of received packets, delivered by this layer to a higher layer, which were addressed to a broadcast address at this layer.

Note that IPv6 does not define broadcast addresses, so IPv6 packets never cause this counter to increment.

UnknownProtoPacketsReceived

The total number of packets received via the interface which were discarded because of an unknown or unsupported protocol.

IPPing:1 Parameters

The IPPing:1 profile provides parameters for running Ping diagnostic tests.

Device.IP.Diagnostics.IPPing Parameters

These parameters provide access to an IP-layer Ping test.

DiagnosticsState

Indicates availability of diagnostic data; one of:

- None
- Requested
- Complete
- Error_CannotResolveHostName
- Error_Internal
- Error_Other

Only Requested may be written, to start the Ping test. When the test completes, the value of this parameter is either Complete (if the test completed successfully), or one of the Error values listed above. If Requested is written while a test is in progress, the test restarts.

To stop a test in progress, write to any of the other Ping parameters. In this situation, the DiagnosticsState is None.

Interface

The path name to a row in the Device.IP.Interface table, specifying the interface used for the Ping test. Example: Device.IP.Interface.1

Use an empty string to specify the interface as directed by the routing policy (Forwarding table entries).

Host

Host name or address of the host to ping.

NumberOfRepetitions

The number of repetitions of the Ping test to perform before reporting the results.

Timeout

The timeout, in milliseconds, for the Ping test.

DataBlockSize

The size of the data block, in bytes, to be sent for each Ping.

DSCP

The DiffServ codepoint to be used for the test packets.

SuccessCount

A result parameter, indicating the number of successful Pings (that is, a successful response was received before the timeout) in the most recent Ping test.

FailureCount

A result parameter, indicating the number of failed Pings in the most recent Ping test.

AverageResponseTime

A result parameter, indicating the average response time in milliseconds over all repetitions with successful responses of the most recent Ping test.

MinimumResponseTime

A result parameter indicating the minimum response time in milliseconds over all repetitions with successful responses of the most recent Ping test.

MaximumResponseTime

A result parameter indicating the maximum response time in milliseconds over all repetitions with successful responses of the most recent Ping test.

MoCA:1 Parameters

The following parameters are available under the MoCA:1 profile. In this profile, the first index is always 1.

Device.MoCA Parameters

The top-level MoCA interface.

InterfaceNumberOfEntries

The number of entries in the Device.MoCA.Interface table.

Device.MoCA.Interface.{i} Parameters

Describes the PHY and MAC-level configuration of a MoCA interface.

Enable

Set to True to enable this entry.

Status

The current MoCA interface operating status; one of:

- Up
- Down
- Unknown
- Dormant
- NotPresent
- LowerLayerDown
- Error

Alias

A handle, used by the ACS to refer to this entry.

Name

The MoCA interface name assigned by the device.

LastChange

The time, in seconds, since the interface entered its current state.

Upstream

False to indicate this interface is LAN-facing.

MACAddress

The MAC address of the MoCA interface.

FirmwareVersion

The MoCA firmware version.

MaxBitRate

The maximum MoCA PHY rate, in Mbps.

HighestVersion

The highest MoCA version supported on this interface.

CurrentVersion

The MoCA version currently in use.

NetworkCoordinator

The Node ID of the current Network Controller (NC) on the MoCA network.

PrivacyEnabledSetting

The configured link-layer privacy mode. This may differ from PrivacyEnabled if set after network formation or admission.

PrivacyEnabled

True is link-layer security is enabled.

FreqCapabilityMask

A hex-encoded 64-bit mask of supported frequencies. The least significant bit corresponds to 800 MHz, and each bit represents roughly 25 MHz of spectrum. Thus, an interface that supports 1150 MHz through 1500 MHz has a value of 0x000000001FFFC000.

FreqCurrentMaskSetting

The configured 64-bit hex-encoded mask of frequencies in use. This may differ from FreqCurrentMask if set after network formation or admission.

FreqCurrentMask

A hex-encoded 64-bit mask of frequencies in use. The bits set in this mask are a subset of those in FreqCapabilityMask.

CurrentOperFreq

The current frequency, in Hz, the MoCA interface is using.

KeyPassphrase

The MoCA password, 12 to 17 numeric characters.

NodeID

The Node ID of this interface.

AssociatedDeviceNumberOfEntries

The number of entries in the Device.MoCA.Interface.{i}.AssociatedDevice table.

Device.MoCA.Interface.{i}.Stats Parameters

Throughput statistics for this MoCA interface.

BytesSent

The total number of bytes transmitted from the interface, including framing characters.

BytesReceived

The total number of bytes received on the interface, including framing characters.

PacketsSent

The total number of packets transmitted from the interface.

PacketsReceived

The total number of packets received on the interface.

Device.MoCA.Interface.{i}.AssociatedDevice.{i} Parameters

This table contains information about other MoCA devices associated with this MoCA interface.

MACAddress

The MAC address of the associated MoCA device.

NodeID

The node ID of the associated device.

PreferredNC

True if the associated device is a preferred Network Controller.

HighestVersion

The highest MoCA version the associated device supports.

PHYTxRate

The PHY transmit rate, in Mbps, to the associated device.

PHYRxRate

The PHY receive rate, in Mbps, from the associated device.

TxPowerControlReduction

The reduction in transmit level, in dB, due to power control.

RxPowerLevel

The power level, in dBm, received at the MoCA interface from the associated device.

TxBcastRate

The broadcast PHY rate, in Mbps, from the associated device.

RxBcastPowerLevel

The broadcast power level, in dBm, received from the associated device.

TxPackets

The number of packets sent to the associated device, including broadcast, multicast, and unicast packets.

RxPackets

The number of packets received from the associated device, including broadcast, multicast, and unicast packets.

RxErroredAndMissedPackets

The number of errored and missed packets received from the associated device.

QAM256Capable

True if the associated device supports 256QAM.

PacketAggregationCapability

The packet aggregation capability supported by the associated device. Standard values are 0 (no support), 6 packets, or 10 packets.

RxSNR

The SNR, in dBm, received from the associated device.

Active

True if the associated device is currently present on the MoCA network.

NAT:1 Parameters

The following parameters are available under the NAT:1 profile.

Device.NAT Parameters

Describes Network Address Translation (NAT) capabilities.

InterfaceSettingNumberOfEntries

The number of entries in the Device.NAT.InterfaceSetting table.

PortMappingNumberOfEntries

The number of entries in the Device.NAT.PortMapping table.

Device.NAT.InterfaceSetting.{i} Parameters

NAT settings for a specific interface.

Enable

Set to True to enable this entry.

Status

The status of this entry; one of:

- Disabled
- Enabled
- Enabled_NATForcedDisabled (NAT enabled, but forced by a third party to be operationally disabled; for example, because a MAP.Domain is enabled but there is no Basic Mapping Rule)
- Enabled_PortMappingDisabled (NAT enabled, but port mapping has been operationally disabled by a third party; for example, the current Firewall level requires NAT to be disabled)
- Error_Misconfigured
- Error

Interface

The path name to a row in the Device.IP.Interface table, indicating the interface this entry applies to.

Device.NAT.PortMapping Parameters

The NAT port mapping table.

Enable

Set to True to enable this entry.

Interface

The path name to a row in the Device.IP.Interface table, indicating the interface this entry applies to.

AllInterfaces

If True, this entry applies to all IP interfaces that support port mapping. The Interface parameter is unused in this case.

LeaseDuration

The time to live, in seconds, for the port mapping lease. Set to 0 for a static port mapping.

RemoteHost

The IP address of the source of inbound packets, or an empty string to indicate any IP address (the usual case).

ExternalPort

The external port that the NAT gateways listens on for traffic to the corresponding InternalPort. Set to 0 to indicate all ports.

InternalPort

The port on the device, indicated by InternalClient, that the gateway should forward traffic to.

Protocol

The port mapping protocol; one of:

- TCP
- UDP

InternalClient

The IP address or DNS host name of an internal client on the LAN.

Description

Text describing this port mapping.

Routing:2 Parameters

The following parameters are available under the Routing:2 profile.

Device.Routing Parameters

Contains the Routing table and RIP configuration.

RouterNumberOfEntries

Number of entries in the Router table.

Device.Routing.RIP Parameters

The Routing Information Protocol (RIP) configuration.

Enable

Set to True to enable RIP.

SupportedModes

The supported RIP protocol modes; one of:

- Send
- Receive
- Both

InterfaceSettingNumberOfEntries

The number of entries in the Device.Routing.RIP.InterfaceSetting table.

Device.Routing.RIP.InterfaceSetting.{i}. Parameters

The RIP configuration table for a given interface.

Enable

Set to True to enable this entry.

Status

The status of this entry; one of:

- Disabled
- Enabled
- Error_Misconfigured
- Error

Interface

A path name to a row in the Device.IP.Interface table, indicating the interface associated with this entry.

Version

The RIP version used on this interface.

AcceptRA

Set to True to accept RIP route advertisements from the interface.

SendRA

Set to True to send RIP route advertisements over this interface.

Device.Routing.Router.{i} Parameters

Configures routing and forwarding.

Enable

Set to True to enable this entry.

Status

The status of this entry; one of:

- Disabled
- Enabled
- Error

IPv4ForwardingNumberOfEntries

The number of entries in the Device.Routing.Router.{i}.IPv4Forwarding table.

IPv6ForwardingNumberOfEntries

The number of entries in the Device.Routing.Router.{i}.IPv6Forwarding table.

Device.Routing.Router.{i}.IPv4Forwarding.{i} Parameters

The Layer 3 IPv4 forwarding table.

Enable

Set to True to enable this entry..

Status

The status of this entry; one of:

- Disabled
- Enabled
- Error_Misconfigured
- Error

DestIPAddress

The destination IPv4 address, or an empty string to indicate no specified address. If DestIPAddress and DestSubnetMask are both empty strings, this is a default route.

DestSubnetMask

The destination subnet mask, or an empty string to indicate no specified subnet.

GatewayIPAddress

The IPv4 address of the gateway.

Interface

A path name to a row in the Device.IP.Interface table, specifying the egress Layer 3 interface associated with this entry.

ForwardingMetric

The forwarding metric value, or -1 to indicate the metric is not used.

StaticRoute

True if this entry defines a Static route.

Origin

The protocol used to learn the IPv4 forwarding rule associated with this entry; one of:

- DHCPv4
- OSPF
- IPCP
- RIP
- Static (default entries, or entries created by the ACS or the web-based interface)

Device.Routing.Router.{i}.IPv6Forwarding.{i} Parameters

The Layer 3 IPv6 forwarding table.

Enable

Set to True to enable this entry.

Status

The status of this entry; one of:

- Disabled
- Enabled
- Error_Misconfigured
- Error

DestIPPrefix

The destination IPv6 prefix. An empty string matches all prefixes.

NextHop

The IPv6 address of the next hop. Set a value for either Interface or NextHop, not both.

Interface

A path name to a row in the Device.IP.Interface table, specifying the egress Layer 3 interface associated with this entry.

ForwardingMetric

The forwarding metric value, or -1 to indicate the metric is not used.

Origin

The protocol used to learn the IPv6 forwarding rule associated with this entry; one of:

- DHCPv6
- OSPF
- RA (Router Advertisement Route Information)
- RIPng
- Static (default entries, or entries created by the ACS or the web-based interface)

Time:1 Parameters

The following parameters are available under the Time:1 profile.

Device.Time Parameters

Provides information about the NTP or SNTP client on the device.

Enable

Set to True to enable the NTP or SNTP client.

Status

The client status; one of:

- Disabled
- Unsynchronized (device time has not been set yet)
- Synchronized
- Error_FailedToSynchronize (failed to acquire time; current time is inaccurate)
- Error

NTPServer1

Host name or IP address of the primary Time server.

NTPServer2

Host name or IP address of the secondary Time server.

CurrentLocalTime

The current date and time in the device's local time zone.

LocalTimeZone

The POSIX local time zone definition.

TraceRoute:1 Parameters

The TraceRoute:1 profile provides parameters for running Traceroute diagnostic tests.

Device.IP.Diagnostics.TraceRoute Parameters

These parameters define access to an IP-layer Traceroute test for the specified IP interface.

DiagnosticsState

Indicates availability of diagnostic data; one of:

- None
- Requested
- Complete
- Error_CannotResolveHostName
- Error_MaxHopCountExceeded

Only Requested may be written, to start the Traceroute test. When the test completes, the value of this parameter is either Complete (if the test completed successfully), or one of the Error values listed above.

Interface

The path name to a row in the Device.IP.Interface table, specifying the interface used for the Traceroute test.

Host

Host name or address of the host to find a route to.

NumberOfTries

The number of tries per hop. Valid range: 1 to 3. Default: 3.

Timeout

The timeout, in milliseconds, for each hop of the Traceroute test. Default: 5000.

DataBlockSize

The size of the data block, in bytes, to be sent for each Traceroute. Valid range: 1 to 65535. Default: 38.

DSCP

The DiffServ codepoint to use for the test packets.

MaxHopCount

The maximum number of hops used in outgoing probe packets (max TTL). Valid range: 1 to 64. Default: 30.

ResponseTime

A result parameter, indicating the response time in milliseconds for the most recent Traceroute test.

RouteHopsNumberOfEntries

The number of entries in the RouteHops table.

Device.IP.Diagnostics.TraceRoute.RouteHops.{i} Table

This table contains the returned hop results. If a route could not be determined, this table is empty.

Host

A result parameter, describing a hop along the discovered route. If DNS can resolve a host name, this is the FQDN of the host. Otherwise, this is the IP address of the hop.

HostAddress

If this parameter is not an empty string, it contains the last IP address of the host returned for this hop; the Host contains the Host Name returned from the reverse DNS query.

ErrorCode

Contains the error code returned for this hop. This code is directly from the ICMP CODE field.

RTTimes

A comma-separated list (maximum list length 16) of unsigned integers. Each item contains one or more round trip times in milliseconds (one for each repetition) for this hop.

Upload:1 Parameters

The following parameters are available under the Upload:1 profile. They support the upload phase of a TR-143 Speedtest.

Device.IP.Diagnostics.UploadDiagnostics Parameters

These parameters define the diagnostics configuration for a HTTP or FTP UploadDiagnostics test.

DiagnosticsState

Indicates the availability of diagnostic data; one of:

- None
- Requested
- Completed
- Error_InitConnectionFailed
- Error_NoResponse
- Error_PasswordRequestFailed
- Error_LoginFailed
- Error_NoTransferMode

- Error_NoPASV
- Error_NoCWD
- Error_NoSTOR
- Error_NoTransferComplete

Only Requested may be written, to start the download test. When the test completes, the value of this parameter is either Completed (if the test completed successfully), or one of the Error values listed above. If Requested is written while a test is in progress, the test restarts.

To stop a test in progress, write to any of the other Download parameters. In this situation, the DiagnosticsState is None.

Interface

The path name to a row in the Device.IP.Interface table, specifying the interface used for the download test. Use an empty string to specify the default routing interface.

UploadURL

An HTTP or FTP URL, identifying the destination resource for the upload. The following considerations apply:

- An FTP upload uses binary mode.
- An HTTP upload uses persistent connections. Pipelining and HTTP Authentication are not used.

UploadTransports

A comma-separated list of strings, identifying supported upload transport protocols for a CPE device. Typically, the string is "HTTP,FTP"

DSCP

The DiffServ code point for marking packets transmitted in the test.

EthernetPriority

The Ethernet priority code for marking packets transmitted in the test (if applicable).

TestFileLength

The size of the file (in bytes) to be uploaded to the server. The device generates enough random data to perform the test.

ROMTime

The request time, in UTC, specified to microsecond precision. For example: 2008-04-09T15:01:05.123456

For HTTP, this is the time when the device sent the PUT command.

For FTP, this is the time when the device sent the STOR command.

BOMTime

The time, in UTC, when transmission began, specified to microsecond precision. For example: 2008-04-09T15:01:05.123456

For HTTP, this is the time when the first data packet was sent.

For FTP, this is the time when the client received the "ready for transfer" notification.

EOMTime

The time, in UTC, when the transmission ended, specified to microsecond precision. For example: 2008-04-09T15:01:05.123456

For HTTP, this is the time when the HTTP "successful" response was received.

For FTP, this is the time when the client received a "transfer complete" notification.

TotalBytesSent

The total number of bytes sent on the Interface between BOMTime and EOMTime.

TCPOpenRequestTime

The request time, in UTC, specified to microsecond precision. For example: 2008-04-09T15:01:05.123456

For HTTP, this is the time when the TCP socket open (SYN) was sent for the HTTP connection.

For FTP, this is the time when the TCP socket open (SYN) was sent for the data connection.

TCPOpenResponseTime

The response time, in UTC, specified to microsecond precision.

For HTTP, this is the time when the TCP ACK to the socket opening the HTTP connection was received.

For FTP, this is the time when the TCP ACK to the socket opening the data connection was received.

UPnPDev:1 Parameters

The following parameters are available under the UPnPDev:1 profile.

Device.UPnP.Device Parameters

Defines the UPnP devices and services available on this device.

Enable

Set to True to enable UPnP support.

UPnPMediaServer

Set to True to enable UPnP Media Server support.

UPnPMediaRenderer

Set to True to enable UPnP Media Renderer support.

UPnPWLANAccessPoint

Set to True to enable UPnP Wireless Access Point support.

UPnPQoSDevice

Set to True to enable UPnP QoS Device support.

UPnPQoSPolicyHolder

Set to True to enable UPnP QoS Policy Holder support.

Device.UPnP.Device.Capabilities Parameters

Lists the UPnP capabilities for this device.

UPnPArchitecture

The major version of supported UPnP architecture.

UPnPMediaServer

The supported revision for UPnP Media Server, or 0 for no support.

UPnPMediaRenderer

The supported revision for UPnP Media Renderer, or 0 for no support.

UPnPWLANAccessPoint

The supported revision for UPnP Wireless Access Point, or 0 for no support.

UPnPBasicDevice

The supported revision for UPnP Basic Device, or 0 for no support.

UPnPQoSDevice

The supported revision for UPnP QoS Device, or 0 for no support.

UPnPQoSPolicyHolder

The supported revision for UPnP QoS Policy Holder, or 0 for no support.

UPnPIGD

The supported revision for UPnP IGD, or 0 for no support.

User:1 Parameters

The following parameters are available under the User:1 profile.

Device.Users

The collection of user accounts on this device.

UserNumberOfEntries

The number of entries in the Device.Users.User table.

Device.Users.User.{i} Parameters

Enable

Set to True to enable this entry.

Username

The user ID for this entry. On Touchstone devices, the standard user names are "admin" and "technician."

Password

The password associated with the user ID.

Language

The default language for this user, or a blank string to use the user interface's current language settings.

RemoteAccessCapable

Set to True to allow remote access to the device for this user.

WiFiAccessPoint:1 Parameters

The following parameters are available under the WiFiAccessPoint:1 profile.

Device.WiFi. Parameters

The top-level Wi-Fi parameters. Defines Radio, SSID, and AccessPoint tables.

AccessPointNumberOfEntries

Number of entries in the Device.WiFi.AccessPoint table.

RadioNumberOfEntries

Number of entries in the Device.WiFi.Radio table.

SSIDNumberOfEntries

Number of entries in the Device.WiFi.SSID table.

Device.WiFi.AccessPoint.{i} Parameters

Provides a model of the Wi-Fi connection from the perspective of the access point.

Enable

Set to True to enable this entry.

Status

Access point status; one of:

- Disabled
- Enabled
- Error_Misconfigured
- Error

Alias

A handle, used by the ACS to refer to this entry..

SSIDReference

The path name to a row in the Device.WiFi.SSID table.

SSIDAdvertisementEnabled

True if beacons advertise the SSID name.

RetryLimit

The maximum number of retransmissions for a packet. Valid range: 0 to 7.

WMMCapability

True if this access point supports Wi-Fi Multimedia (WMM) Access Categories.

UAPSDCapability

True if this access point supports WMM Unscheduled Automatic Power Save Delivery (UAPSD).

WMMEnable

True is WMM support is enabled.

UAPSEnable

True if UAPSD support is enabled.

X_ARRIS_COM_RadiusOperatorName

The RADIUS Attribute operator name.

X_ARRIS_COM_RadiusLocationInformation

The RADIUS Attribute location information.

X_ARRIS_COM_RadiusLocationData

The RADIUS Attribute location data.

AssociatedDeviceNumberOfEntries

The number of entries in the Device.WiFi.AccessPoint.{i}.AssociatedDevice table.

Device.WiFi.AccessPoint.{i}.Security Parameters

Contains security-related parameters that apply to a device acting as an Access Point.

ModesSupported

A comma-separated list of strings, indicating the supported security modes. One or more of:

- None
- WEP-64
- WEP-128
- WPA-Personal
- WPA2-Personal
- WPA-WPA2-Personal
- WPA-Enterprise
- WPA2-Enterprise
- WPA-WPA2-Enterprise

ModeEnabled

Indicates which supported security mode is in effect.

WEPKey

A hexadecimal string representing a WEP key. Used only for WEP-64 and WEP-128 security modes.

PreSharedKey

A hexadecimal string representing a Pre-Shared Key (PSK). Used only for WPA-Personal, WPA2-Personal, or WPA-WPA2-Personal security modes.

KeyPassphrase

A passphrase used to generate the pre-shared key. Used only for WPA-Personal, WPA2-Personal, or WPA-WPA2-Personal security modes.

RekeyingInterval

The interval, in seconds, in which keys are regenerated. Applies to all WPA and WPA2 modes.

RadiusServerIPAddr

The IP address of the RADIUS server used for WLAN security. Applies only to WPA or WPA2 Enterprise modes.

RadiusServerPort

The port number of the RADIUS server.

RadiusSecret

The secret used for handshaking with the RADIUS server.

SecondaryRadiusServerIPAddr

The internet address of the Secondary RADIUS server for this entry.

SecondaryRadiusServerPort

The UDP port used to communicate with the Secondary RADIUS server for this entry.

SecondaryRadiusServerSecret

The Secondary RADIUS key for this entry.

X_ARRIS_COM_EncryptionMode

The WPA encryption mode used by this entry.

Device.WiFi.AccessPoint.{i}.WPS Parameters

Provides parameters related to Wi-Fi Protected Setup (WPS) for this access point.

Enable

True if WPS is enabled on this access point.

ConfigMethodsSupported

A comma-separated list of strings, indicating supported WPS configuration methods. One or more of:

- USBFlashDrive
- Ethernet
- ExternalNFCToken
- IntegratedNFCToken
- NFCInterface
- PushButton
- PIN

ConfigMethodsEnabled

A comma-separated list of strings, indicating WPS configuration methods actually in use.

Device.WiFi.AccessPoint.{i}.AssociatedDevice.{i} Parameters

This table lists devices currently associated with the access point.

MACAddress

The MAC address of the associated device.

AuthenticationState

True if the associated device has authenticated.

LastDataDownlinkRate

The data transmit rate, in kbps, that was most recently used for transmission from the Access Point to the associated device.

LastDataUplinkRate

The data transmit rate, in kbps, that was most recently used for transmission from the associated device to the Access Point.

SignalStrength

The radio signal strength of the uplink, from the associated device to the access point, measured in dBm, as an average of the last 100 packets received from the device.

Retransmissions

The number of packets that were re-transmitted, of the last 100 packets sent to the associated device. Multiple re-transmissions of the same packet count as one.

Active

True if this node is currently present in the Wi-Fi AccessPoint network.

MACAddressControlEnabled

Set to All to allow all devices to connect, Allow to allow only AllowedMACAddress entries to connect, or Deny to deny all AllowedMACAddress entries.

AllowedMACAddress

The MAC address of a device that is allowed to connect if MACAddressControlEnabled is Allow, or denied if MACAddressControlEnabled is Deny.

Device.WiFi.AccessPoint.{i}.NeighboringWiFiDiagnostic Parameters

Defines access to other Wi-Fi SSIDs that this device is able to receive.

DiagnosticsState

The availability of Wi-Fi SSID data; one of:

- None
- Requested
- Completed
- Error

To begin a Wi-Fi scan, set this parameter to Requested. Other values are read-only. Writing this value while a test is in progress restarts the scan.

ResultNumberOfEntries

The number of entries in the
Device.WiFi.AccessPoint{i}.NeighboringWiFiDiagnostic.Result table.

Device.WiFi.AccessPoint{i}.NeighboringWiFiDiagnostic.Result.{i} Parameters

Contains results of the Wi-Fi scan.

Radio

The path name to a device in the Device.WiFi.Radio table, indicating the radio that detected the neighboring SSID.

SSID

The current SSID in use by the neighboring SSID, or empty for hidden SSIDs.

BSSID

The BSSID used for the neighboring SSID.

Channel

The radio channel used by the neighboring Wi-Fi radio.

SignalStrength

The Radio Signal Strength (RSSI) of the neighboring Wi-Fi radio, measured in dBm, measured over the last 100 packets received.

OperatingFrequencyBand

The frequency band the SSID is using; one of:

- 2.4GHz
- 5GHz

SupportedStandards

A comma-separated list, indicating which 802.11 standards the result SSID can support in the specified FrequencyBand. One or more of:

- a
- b
- g
- n
- ac

Noise

The average noise strength, in dBm, received from the neighboring Wi-Fi radio.

SupportedDataTransferRates

A comma-separated list of strings, indicating data transmit rates in Mbps at which the SSID permits stations to connect.

OperatingStandards

A comma-separated list of strings, representing a subset of SupportedStandards, indicating the standards actually supported by the radio associated with the result SSID. For example, a 5 GHz radio only allows one or more of "a,n,ac."

SecurityModeEnabled

The encryption type advertised by the neighboring SSID. One of:

- None
- WEP-64
- WEP-128
- WPA-Personal
- WPA2-Personal
- WPA-WPA2-Personal
- WPA-Enterprise
- WPA2-Enterprise
- WPA-WPA2-Enterprise

OperatingChannelBandwidth

The bandwidth the channel is using. One of:

- 20MHz
- 40MHz
- 80MHz
- 160MHz
- Auto

Mode

The mode the neighboring Wi-Fi radio is using; one of:

- AdHoc
- Infrastructure

EncryptionMode

A comma-separated list of strings, indicating advertised encryption modes. One or more of:

- TKIP
- AES

BeaconPeriod

The interval, in milliseconds, between transmitting beacons.

BasicDataTransferRate

A comma-separated list of strings, indicating basic data transmit rates (in Mbps) for the SSID. For example, if BasicDataTransferRates is "1,2", this indicates that the SSID is operating with basic rates of 1 Mbps and 2 Mbps.

DTIMPeriod

The number of beacon intervals that elapse between transmission of Beacon frames, containing a TIM element whose DTIM count field is 0. This value is transmitted in the DTIM Period field of beacon frames.

Device.WiFi.AccessPoint.{i}.Accounting Parameters

Provides advanced RADIUS configuration information.

ServerIPAddr

The IP address of the primary Accounting server for this entry.

SecondaryServerIPAddr

The IP address of a secondary Accounting server for this entry.

X_ARRIS_COM_SecondaryServerIpAddr2

The IP address of a third Accounting server for this entry.

X_ARRIS_COM_SecondaryServerIpAddr3

The IP address of a fourth Accounting server for this entry.

ServerPort

The UDP port used to communicate with the primary Accounting server.

SecondaryServerPort

The UDP port used to communicate with the secondary Accounting server.

X_ARRIS_COM_SecondaryServerPort2

The UDP port used to communicate with the third Accounting server.

X_ARRIS_COM_SecondaryServerPort3

The UDP port used to communicate with the fourth Accounting server.

Secret

The Accounting Server Authentication key for the primary Accounting server.

SecondarySecret

The Accounting Server Authentication key for the secondary Accounting server.

X_ARRIS_COM_SecondarySecret2

The Accounting Server Authentication key for the third Accounting server.

X_ARRIS_COM_SecondarySecret3

The Accounting Server Authentication key for the fourth Accounting server.

InterimInterval

The interval, in seconds, between report generation. Valid range: 0 to 28800. Default: 600.



Note: To avoid network traffic congestion, ARRIS recommends a minimum interval of 600 seconds.

WiFiRadio:1 Parameters

The following parameters are available under the WiFiRadio:1 profile. In this profile,

Device.WiFi. Parameters

Points to the device radios.

RadioNumberOfEntries

The number of entries in the Device.WiFi.Radio table.

Device.WiFi.Radio.{i} Parameters

These parameters describe individual radios in the device. The index is the [ifIndex](#) of the radio (10000 for the 2.4GHz radio, 10100 for the 5.0GHz radio).

Enable

Set to True to enable this radio.

Status

The current operational state; one of:

- Up
- Down
- Unknown
- Dormant
- NotPresent
- LowerLayerDown
- Error

Alias

A handle, used by the ACS to refer to this entry.

Name

The name of this radio.

Upstream

True if this radio points toward the WAN, False if it points to LAN devices.

MaxBitRate

The maximum PHY bit rate for this radio, in Mbps.

SupportedFrequencyBands

A comma-separated list of strings, indicating the frequency bands at which this radio can operate. One or more of:

- 2.4GHz
- 5GHz

OperatingFrequencyBand

The frequency band this radio is using.

- "2.4GHz" for Device.WiFi.Radio.10000
- "5GHz" for Device.WiFi.Radio.10100

SupportedStandards

A comma-separated list of strings, indicating the 802.11 standards supported by this radio.

- "b,g,n" for Device.WiFi.Radio.10000
- "a,n,ac" for Device.WiFi.Radio.10100

OperatingStandards

The actual 802.11 standards configured for use by this radio. A subset of SupportedStandards.

RegulatoryDomain

A string, composed of a two-letter country code and one of the following characters:

- space: all environments
- I: inside environments
- O: outside environments

Specifying a country code restricts channels and power levels as needed to conform to that country's regulations. The "DF" (default) code is the most restrictive, and is legal in most countries.

PossibleChannels

A comma-separated list of strings, representing the channels allowed for the configured standard (a, b, g, n, ac) and the configured RegulatoryDomain.

Channel

The current radio channel in use. If AutoChannelEnable is True, this parameter represents the automatically-selected channel.

AutoChannelSupported

True if the radio supports automatic channel selection.

AutoChannelEnable

Set to True to enable automatic channel selection.

TransmitPowerSupported

A comma-separated list of integers, representing supported transmit power levels as a percentage of full power. Default: "25,50,75,100"

TransmitPower

The transmit power in use. One of the choices listed by TransmitPowerSupported.

ExtensionChannel

The secondary extension channel position, when operating in wide channel mode (40 MHz or greater).

GuardInterval

The guard interval between OFDM symbols (802.11n connections only). One of:

- 400nsec

- 800nsec
- Auto

MCS

For 802.11n connections, the Modulation Coding Scheme (MCS) index. A value of -1 indicates automatic MCS index selection.

IEEE80211hSupported

True if 802.11h functionality is supported by this radio.

IEEE80211hEnabled

Set to true to enable 802.11h functionality (only if supported).

ChannelsInUse

A comma-separated list of strings, indicating the channels the radio detects to be in use, including the channel(s) it is using.

LowerLayers

(unused) A comma-separated list of path names to interfaces stacked immediately below this interface.

BeaconPeriod

The time, in milliseconds, between beacon transmissions.

Device.WiFi.Radio.{i}.Stats Parameters

Throughput statistics for this radio.

BytesSent

The total number of bytes transmitted from this radio, including framing characters.

BytesReceived

The total number of bytes received on this radio, including framing characters.

PacketsSent

The total number of packets transmitted from this radio.

PacketsReceived

The total number of packets received by this radio.

ErrorsSent

The total number of packets that could not be transmitted from this radio due to errors.

ErrorsReceived

The total number of received packets that were discarded due to errors.

DiscardPacketsSent

The total number of outbound packets discarded, although no errors were detected (for example, due to buffer overflow).

DiscardPacketsReceived

The total number of received packets discarded, although no errors were detected (for example, due to buffer overflow).

WiFiSSID:1 Parameters

The following parameters are available under the WiFiSSID:1 profile. In this profile, the index is the **ifindex** of the SSID (10001-10008 for the 2.4GHz SSIDs and 10101-10108 for the 5.0GHz SSIDs).

Device.WiFi. Parameters

Points to the device radios.

SSIDNumberOfEntries

Number of entries in the Device.WiFi.SSID table.

Device.WiFi.SSID.{i} Parameters

The Wi-Fi SSID table.

Enable

Set to True to enable this SSID entry.

Status

The current state of the entry; one of:

- Up
- Down
- Unknown
- Dormant
- NotPresent
- LowerLayerDown
- Error

Alias

A handle, used by the ACS to refer to this entry.

Name

The name of this SSID, assigned by the device.

LastChange

The time, in seconds, since the operating status last changed.

LowerLayers

A comma-separated list of path names to interface objects stacked directly below this interface.

BSSID

The Basic Service Set ID. The MAC address of the access point; local to model an access point SSID or remote to model an endpoint SSID.

MACAddress

The MAC address of this interface.

SSID

The SSID name in use by the connection.

Device.WiFi.SSID.{i}.Stats Parameters

Throughput statistics for this SSID.

BytesSent

The total number of bytes transmitted from the interface, including framing characters.

BytesReceived

The total number of bytes received on the interface, including framing characters.

PacketsSent

The total number of packets transmitted from the interface.

PacketsReceived

The total number of packets received on the interface.

ErrorsSent

The total number of outbound packets that could not be transmitted because of errors.

ErrorsReceived

The total number of inbound packets that contained errors, preventing them from being delivered to a higher-layer protocol.

UnicastPacketsSent

The total number of packets requested for transmission which were not addressed to a multicast or broadcast address at this layer, including those that were discarded or not sent.

UnicastPacketsReceived

The total number of received packets, delivered by this layer to a higher layer, which were not addressed to a multicast or broadcast address at this layer.

DiscardPacketsSent

The total number of outbound packets which were discarded even though no errors had been detected to prevent their being transmitted. (For example, to free up buffer space.)

DiscardPacketsReceived

The total number of inbound packets which were discarded even though no errors had been detected to prevent their being delivered. (For example, to free up buffer space.)

MulticastPacketsSent

The total number of packets that higher-level protocols requested for transmission and which were addressed to a multicast address at this layer, including those that were discarded or not sent.

MulticastPacketsReceived

The total number of received packets, delivered by this layer to a higher layer, which were addressed to a multicast address at this layer.

BroadcastPacketsSent

The total number of packets that higher-level protocols requested for transmission and which were addressed to a broadcast address at this layer, including those that were discarded or not sent.

BroadcastPacketsReceived

The total number of received packets, delivered by this layer to a higher layer, which were addressed to a broadcast address at this layer.

UnknownProtoPacketsReceived

The total number of packets received via the interface which were discarded because of an unknown or unsupported protocol.

ArrisFirewall Parameters

The parameters in this profile are part of the Device.X_ARRIS_COM_Firewall block, and provide the access controls available in the Gateway WebGUI.

Device.X_ARRIS_COM_Firewall.ParentalControls Parameters

These parameters configure and manage Parental Controls.

Enable

Set to true to enable Parental Controls.

TrustedMACNumberOfEntries

The number of entries in the Device.X_ARRIS_COM_Firewall.ParentalControls.TrustedMAC table.

KeywordFilterNumberOfEntries

The number of entries in the Device.X_ARRIS_COM_Firewall.ParentalControls.KeywordFilter table.

WebSiteBlackListedNumberOfEntries

The number of entries in the Device.X_ARRIS_COM_Firewall.ParentalControls.WebSiteDenyList table.

WebSiteWhiteListedNumberOfEntries

The number of entries in the Device.X_ARRIS_COM_Firewall.ParentalControls.

WebSiteListSelect

Set to 1 to use the blacklist, or 2 to use the whitelist.

Device.X_ARRIS_COM_Firewall.ParentalControls.TrustedMAC.{i} Parameters

These parameters define up to two MAC addresses whose computers are not affected by parental controls.

Enable

Set to True to enable this entry.

MACAddress

The MAC address of a trusted computer on the LAN.

Device.X_ARRIS_COM_Firewall.ParentalControls.KeywordFilter.{i}

Parameters

Rows in this table define keywords that block URLs with matching substrings.

Enable

Set to True to enable this entry.

Keyword

The keyword to block.

DayNumberOfEntries

The number of entries in the

Device.X_ARRIS_COM_Firewall.ParentalControls.KeywordFilter.{i}.Day table.

Device.X_ARRIS_COM_Firewall.ParentalControls.KeywordFilter.{i}.Day.{i}

Parameters

Rows in this table define days when keyword filtering is in effect. Valid index values are 1 through 7, where 1 represents Sunday and 7 represents Saturday.

Enable

Set to True to enable this entry.

TimeNumberOfEntries

The number of entries in the

Device.X_ARRIS_COM_Firewall.ParentalControls.KeywordFilter.{i}.Time table.

Device.X_ARRIS_COM_Firewall.ParentalControls.KeywordFilter.{i}.Time.{i}

Parameters

Rows in this table define the starting and ending hours that keyword filtering is in effect.

Enable

Set to True to enable this entry.

StartHour

The hour at which keyword filtering begins. Valid range: 0 (midnight) to 23 (11 p.m.).

EndHour

The hour at which keyword filtering ends.

Device.X_ARRIS_COM_Firewall.ParentalControls.WebSiteDenyList.{i}

Parameters

This table defines blacklisted web sites.

Enable

Set to True to enable this entry.

WebSite

The domain name of the web site to block.

DayNumberOfEntries

The number of entries in the

Device.X_ARRIS_COM_Firewall.ParentalControls.WebSiteDenyList.{i}.Day table.

Device.X_ARRIS_COM_Firewall.ParentalControls.WebSiteDenyList.{i}.Day.{i} Parameters

Rows in this table represent days web site filtering is in effect. Valid index values are 1 through 7, where 1 represents Sunday and 7 represents Saturday.

Enable

Set to True to enable this entry.

TimeNumberOfEntries

The number of entries in the

Device.X_ARRIS_COM_Firewall.ParentalControls.WebSiteDenyList.{i}.Day.{i}.Time table.

Device.X_ARRIS_COM_Firewall.ParentalControls.WebSiteDenyList.{i}.Day.{i}.Time.{i} Parameters

Rows in this table define the starting and ending hours that web site filtering is in effect.

Enable

Set to True to enable this entry.

StartHour

The hour at which web site filtering begins. Valid range: 0 (midnight) to 23 (11 p.m.).

EndHour

The hour at which web site filtering ends.

Device.X_ARRIS_COM_Firewall.ParentalControls.WebSiteAllowList.{i} Parameters

This table configures a whitelist, web sites that LAN devices may always connect to.

Enable

Set to True to enable this entry.

WebSite

The domain name of the web site to allow.

DayNumberOfEntries

The number of entries in the

Device.X_ARRIS_COM_Firewall.ParentalControls.WebSiteAllowList.{i}.Day table.

Device.X_ARRIS_COM_Firewall.ParentalControls.WebSiteAllowList.{i}.Day.{i} Parameters

Rows in this table represent days whitelisting is in effect. Valid index values are 1 through 7, where 1 represents Sunday and 7 represents Saturday.

Enable

Set to True to enable this entry.

TimeNumberOfEntries

The number of entries in the

Device.X_ARRIS_COM_Firewall.ParentalControls.WebSiteAllowList.{i}.Day.{i}.Time table.

Device.X_ARRIS_COM_Firewall.ParentalControls.WebSiteAllowList.{i}.Day.{i}.Time.{i} Parameters

Rows in this table define the starting and ending hours that whitelisting is in effect.

Enable

Set to True to enable this entry..

StartHour

The hour at which whitelisting begins. Valid range: 0 (midnight) to 23 (11 p.m.).

EndHour

The hour at which whitelisting ends.

Device.X_ARRIS_COM_Firewall.MACFilter Parameters

This table defines MAC addresses that are denied access to the gateway.

MacFilterListSelect

Set to 1 to activate the Deny list, or 2 to activate the Allow list.

DenyListNumberOfEntries

The number of entries in the Device.X_ARRIS_COM_Firewall.MACFilter.DenyList table.

AllowListNumberOfEntries

The number of entries in the Device.X_ARRIS_COM_Firewall.MACFilter.AllowList table.

Device.X_ARRIS_COM_Firewall.MACFilter.DenyList.{i} Parameters

When active, entries in this table are denied access to the network.

Enable

Set to true to enable this entry.

MACAddress

The MAC address to block.

DayNumberOfEntries

The number of entries in the Device.X_ARRIS_COM_Firewall.MACFilter.DenyList.{i}.Day table.

Device.X_ARRIS_COM_Firewall.MACFilter.DenyList.{i}.Day.{i} Parameters

Rows in this table represent days MAC address filtering is in effect. Valid index values are 1 through 7, where 1 represents Sunday and 7 represents Saturday.

Enable

Set to true to enable MAC filtering for this day.

TimeNumberOfEntries

The number of entries in the

Device.X_ARRIS_COM_Firewall.MACFilter.DenyList.{i}.Day.{i}.Time table.

Device.X_ARRIS_COM_Firewall.MACFilter.DenyList.{i}.Day.{i}.Time.{i}**Parameters**

Rows in this table define the starting and ending hours that MAC address filtering is in effect.

Enable

Set to true to enable this row.

StartHour

The hour at which MAC address filtering begins. Valid range: 0 (midnight) to 23 (11 p.m.).

EndHour

The hour at which MAC address filtering ends.

Device.X_ARRIS_COM_Firewall.MACFilter.AllowList.{i} Parameters

When active, entries in this table are allowed access to the network. MAC addresses not represented in this table are denied access.

Enable

Set to True to enable this entry.

MACAddress

The MAC address to allow.

DayNumberOfEntries

The number of entries in the Device.X_ARRIS_COM_Firewall.MACFilter.AllowList.{i}.Day table.

Device.X_ARRIS_COM_Firewall.MACFilter.AllowList.{i}.Day.{i} Parameters

Rows in this table represent days MAC address filtering is in effect. Valid index values are 1 through 7, where 1 represents Sunday and 7 represents Saturday.

Enable

Set to True to enable this entry.

TimeNumberOfEntries

The number of entries in the

Device.X_ARRIS_COM_Firewall.MACFilter.AllowList.{i}.Day.{i}.Time table.

Device.X_ARRIS_COM_Firewall.MACFilter.AllowList.{i}.Day.{i}.Time.{i}**Parameters**

Rows in this table define the starting and ending hours that MAC address filtering is in effect.

Enable

Set to True to enable this entry.

StartHour

The hour at which MAC address filtering begins. Valid range: 0 (midnight) to 23 (11 p.m.).

EndHour

The hour at which MAC address filtering ends.

Device.X_ARRIS_COM_Firewall.IPFILTER Parameters

Defines rules for restricting access to the entire Internet, or selected network services, at specific days and times.

RuleNumberOfEntries

The number of entries in the Device.X_ARRIS_COM_Firewall.IPFILTER.Rule table.

Device.X_ARRIS_COM_Firewall.IPFILTER.Rule.{i} Parameters

Entries in this table define IP filtering rules.

Enable

Set to True to enable this entry.

PortStart

The low port in a range of ports to filter.

PortEnd

The high port in the range. Set PortStart and PortEnd to the same value to filter a single port.

Protocol

The protocol type to filter: UDP, TCP, or BOTH.

Direction

Set to 1 to filter incoming packets, or 2 to filter outgoing packets.

Action

The action to take when the filter matches: 1 to allow, or 2 to deny.

IPAddressType

For IP address filtering, the IP address type to filter: 4 (IPv4) or 6 (IPv6).

IPAddressStart

The low address in a range of IP addresses to filter.

IPAddressEnd

The high address in the range. Set IPAddressStart and IPAddressEnd to the same value to filter a single IP address.

DayNumberOfEntries

The number of entries in the Device.X_ARRIS_COM_Firewall.IPFILTER.Rule.{i}.Day table.

Device.X_ARRIS_COM_Firewall.IPFilter.Rule.{i}.Day Parameters

Rows in this table represent days IP filtering is in effect. Valid index values are 1 through 7, where 1 represents Sunday and 7 represents Saturday.

Enable

Set to True to enable this entry.

TimeNumberOfEntries

The number of entries in the Device.X_ARRIS_COM_Firewall.IPFilter.Rule.{i}.Day.{i}.Time table.

Device.X_ARRIS_COM_Firewall.IPFilter.Rule.{i}.Day.{i}.Time.{i} Parameters

Rows in this table define the starting and ending hours that IP filtering is in effect.

Enable

Set to True to enable this entry.

StartHour

The hour at which IP filtering begins. Valid range: 0 (midnight) to 23 (11 p.m.).

EndHour

The hour at which IP filtering ends.

Band Steering Parameters (AR01.1)

The parameters in this profile are part of the Device.Wifi.X_ARRIS_COM_Bandsteering block.

Capability

A comma-delimited list of supported band steering modes, or "None" if the device does not support Band Steering.

Mode

The current Band Steering mode. In this release, the default mode is "RSSIThresholdAdvanced."

SSIDNumberOfEntries

The number of entries in the Device.WiFi.X_ARRIS_COM_Bandsteering.SSID table.

Device.WiFi.X_ARRIS_COM_Bandsteering.SSID.{i} Parameters

These parameters provide Band Steering settings per SSID. The index is the [ifIndex](#) of the SSID.

Enable

Set to True to enable Band Steering for this SSID.

Active

True if Band Steering is active. Band Steering may be enabled (using the Enable parameter), but remain inactive in any of the following conditions:

- One of the radios is disabled or inactive

- The SSID is disabled or inactive on one or both bands
- The SSID names do not match across bands
- The SSID security settings do not match across bands
- One or more required parameters are not configured

RSSIThreshold

The RSSI threshold, in dBm, for adding or removing a device from the 2.4 GHz temporary blacklist table upon receiving a 5 GHz probe request.

Valid range: -100 to -1. Default: -70

DeltaThreshold

The delta (in dBm) used to compare the RSSI of the probe request on the 5 GHz radio of an associated client.

Valid range: 0 to 100. Default: 5

BlacklistTimeout

The time, in milliseconds, before an entry in the blacklist table expires.

Valid range: 0 to 100000. Default: 15000

ClearCapable5G

Set to True to clear the Device.WiFi.X_ARRIS_COM_Bandsteering.SSID.{i}.5GHzCapable table.

Clear24GHzBlacklist

Set to True to clear the Device.WiFi.X_ARRIS_COM_Bandsteering.SSID.{i}.24GHzBlacklist table.

Capable5GNumberOfEntries

The number of entries in the 5GHzCapable table.

24GHzBlacklistNumberOfEntries

The number of entries in the 24GhzBlacklist table.

Device.WiFi.X_ARRIS_COM_Bandsteering.SSID.{i}.5GHzCapable.{i}**Parameters**

This table lists all 5 GHz-capable clients connected to the SSID.

MACAddress

The client MAC address.

EntryTime

The timestamp, showing when this client was added to the list.

Device.WiFi.X_ARRIS_COM_Bandsteering.SSID.{i}.24GHzBlacklist.{i}**Parameters**

This table lists all clients on the SSID, that have been blacklisted from joining the 2.4 GHz radio.

MACAddress

The MAC Address of the client.

EntryTime

The timestamp, showing when this client was added to the list.

TimeRemaining

The time, in milliseconds, until this entry is removed from the blacklist.

Diagnostics Parameters

The following parameters support performance diagnostics.

Device.Capabilities.PerformanceDiagnostic Parameters

The capabilities of the Performance Diagnostics (DownloadDiagnostics and UploadDiagnostics) for the device.

DownloadTransports

A comma-separated list of strings, listing supported DownloadDiagnostics transport protocols for a CPE device. The list can contain one or more of:

- HTTP
- FTP

UploadTransports

A comma-separated list of strings, listing supported UploadDiagnostics transport protocols for a CPE device. The list can contain one or more of:

- HTTP
- FTP

Device.LAN.IPPingDiagnostics Paramters

These parameters define access to an IP-layer ping test for the default IP interface.

DiagnosticsState

Indicates availability of diagnostic data; one of:

- None
- Requested
- Complete
- Error_CannotResolveHostName
- Error_Internal
- Error_Other

To start the diagnostic test, set this parameter to Requested (only after setting other test parameters to appropriate values). To interrupt a test in progress, write to any of the other test parameters.

When the test is completed, the value of this parameter is either Complete (if the test completed successfully), or one of the Error values. If the value of this parameter is

anything other than Complete, the values of the results parameters for this test are indeterminate. The Gateway retains the results until the next diagnostics test or reboot.

When the diagnostic initiated by the ACS is completed (successfully or not), the CPE establishes a new connection to the ACS, indicating the Event code 8 DIAGNOSTICS COMPLETE in the Inform message.

Host

The FQDN or IP address of the host to ping.

NumberOfRepetitions

The number of pings to send before reporting the results.

Timeout

The timeout, in milliseconds, for each ping.

DataBlockSize

The size of the data block, in bytes, to be sent for each ping.

DSCP

The DiffServ codepoint to be used for the test packets. Default: zero.

SuccessCount

Result parameter, indicating the number of successful pings (those in which a successful response was received prior to the timeout) in the most recent ping test.

FailureCount

Result parameter, indicating the number of failed pings in the most recent ping test.

AverageResponseTime

Result parameter, indicating the average response time in milliseconds over all repetitions with successful responses of the most recent ping test. If there were no successful responses, this value MUST be zero.

MinimumResponseTime

Result parameter, indicating the minimum response time in milliseconds over all repetitions with successful responses of the most recent ping test. If there were no successful responses, this value MUST be zero.

MaximumResponseTime

Result parameter, indicating the maximum response time in milliseconds over all repetitions with successful responses of the most recent ping test. If there were no successful responses, this value MUST be zero.

Device.LAN.TraceRouteDiagnostics Parameters

These parameters define access to an IP-layer trace-route test for the default IP interface.

DiagnosticsState

Indicates availability of diagnostic data; one of:

- None
- Requested

- Complete
- Error_CannotResolveHostName
- Error_MaxHopCountExceeded
- Error_Internal
- Error_Other

To start the diagnostic test, set this parameter to Requested (only after setting other test parameters to appropriate values). To interrupt a test in progress, write to any of the other test parameters.

When the test is completed, the value of this parameter is either Complete (if the test completed successfully), or one of the Error values. If the value of this parameter is anything other than Complete, the values of the results parameters for this test are indeterminate. The Gateway retains the results until the next diagnostics test or reboot.

When the diagnostic initiated by the ACS is completed (successfully or not), the CPE establishes a new connection to the ACS, indicating the Event code 8 DIAGNOSTICS COMPLETE in the Inform message.

Host

The FQDN or IP address of the host to find a route to.

Timeout

The timeout, in milliseconds, for the trace route test.

DataBlockSize

The size of the data block, in bytes, to be sent for each trace route.

MaxHopCount

The maximum number of hop used in outgoing probe packets (max TTL). The default is 30 hops.

DSCP

The DiffServ codepoint to be used for the test packets. Default: zero.

ResponseTime

Result parameter, indicating the response time in milliseconds the most recent trace route test. If a route could not be determined, this value MUST be zero.

NumberOfRouteHops

Result parameter, indicating the number of hops within the discovered route. If a route could not be determined, this value MUST be zero.

Device.LAN.TraceRouteDiagnostics.RouteHops.{i}.

Result parameters, indicating the components of the discovered route. If a route could not be determined, this table is empty. The table consists of the following parameters:

HopHost

Result parameter, indicating the Host Name or IP Address of a hop along the discovered route.

Device.DownloadDiagnostics Parameters

This object defines the diagnostics configuration for a HTTP and FTP DownloadDiagnostics Test.

Files received during DownloadDiagnostics do not require file storage.

DiagnosticsState

Indicates availability of diagnostic data; one of:

- None
- Requested
- Complete
- Error_InitConnectionFailed
- Error_NoResponse
- Error_TransferFailed
- Error_PasswordRequestFailed
- Error_LoginFailed
- Error_NoTransferMode
- Error_NoPASV
- Error_IncorrectSize
- Error_Timeout

To start the diagnostic test, set this parameter to Requested (only after setting other test parameters to appropriate values). To interrupt a test in progress, write to any of the other test parameters.

When the test is completed, the value of this parameter is either Complete (if the test completed successfully), or one of the Error values. If the value of this parameter is anything other than Complete, the values of the results parameters for this test are indeterminate. The Gateway retains the results until the next diagnostics test or reboot.

When the diagnostic initiated by the ACS is completed (successfully or not), the CPE establishes a new connection to the ACS, indicating the Event code 8 DIAGNOSTICS COMPLETE in the Inform message.

Interface

The path name of the IP-layer interface over which the test is to be performed, or an empty string to indicate the default routing interface.

DownloadURL

A URL, indicating the file to download. This is either an HTTP or FTP URL.

DSCP

The DiffServ code point for marking packets transmitted in the test. Default: zero.

EthernetPriority

The Ethernet priority code for marking packets transmitted in the test (if applicable). Default: zero.

ROMTime

The request time in UTC, specified to microsecond precision. For example: 2017-11-26T17:22:05.123456

- For HTTP, this is when the client sent the GET command.
- For FTP, this is when the client sent the RTRV command.

BOMTime

The beginning of transmission time in UTC, specified to microsecond precision.

- For HTTP, this is when the first data packet is received.
- For FTP, this is when the client received the first data packet on the data connection.

EOMTime

The end of transmission in UTC, specified to microsecond precision.

- For HTTP, this is when the last data packet is received.
- For FTP, this is when the client received the last packet on the data connection.

TestBytesReceived

The test traffic received, in bytes, during the download test, including FTP/HTTP headers, between BOMTime and EOMTime,

TotalBytesReceived

The total number of bytes received on the interface between BOMTime and EOMTime.

TCPOpenRequestTime

Request time in UTC, specified to microsecond precision.

- For HTTP, this is when the TCP socket open (SYN) was sent for the HTTP connection.
- For FTP, this is when the TCP socket open (SYN) was sent for the data connection.

TCPOpenResponseTime

Response time in UTC, specified to microsecond precision.

- For HTTP, this is when the TCP ACK to the socket opening the HTTP connection was received.
- For FTP, this is when the TCP ACK to the socket opening the data connection was received.

Device.UploadDiagnostics Parameters

This object defines the diagnostics configuration for a HTTP or FTP UploadDiagnostics test.

Files sent by UploadDiagnostics are a stream of random data.

DiagnosticsState

Indicates availability of diagnostic data; one of:

- None
- Requested
- Complete
- Error_InitConnectionFailed
- Error_NoResponse

- Error_TransferFailed
- Error_PasswordRequestFailed
- Error_LoginFailed
- Error_NoTransferMode
- Error_NoPASV
- Error_NoCWD
- Error_NoSTOR
- Error_NoTransferComplete

To start the diagnostic test, set this parameter to Requested (only after setting other test parameters to appropriate values). To interrupt a test in progress, write to any of the other test parameters.

When the test is completed, the value of this parameter is either Complete (if the test completed successfully), or one of the Error values. If the value of this parameter is anything other than Complete, the values of the results parameters for this test are indeterminate. The Gateway retains the results until the next diagnostics test or reboot.

When the diagnostic initiated by the ACS is completed (successfully or not), the CPE establishes a new connection to the ACS, indicating the Event code 8 DIAGNOSTICS COMPLETE in the Inform message.

Interface

The path name of the IP-layer interface over which the test is to be performed, or an empty string to indicate the default routing interface.

UploadURL

The URL to upload to. This is either an HTTP or FTP URL.

DSCP

The DiffServ code point for marking packets transmitted in the test. Default: zero.

EthernetPriority

The Ethernet priority code for marking packets transmitted in the test (if applicable). Default: zero.

TestFileLength

The size of the file (in bytes) to be uploaded to the server.

ROMTime

The request time in UTC, specified to microsecond precision. For example: 2017-11-26T17:22:05.123456

- For HTTP, this is when the client sends the PUT command.
- For FTP, this is when the STOR command is sent.

BOMTime

The beginning of transmission time in UTC, specified to microsecond precision.

- For HTTP, this is when the first data packet is sent.
- For FTP, this is when the client receives the ready for transfer notification.

EOMTime

End of transmission in UTC, specified to microsecond precision.

- For HTTP, this is when the HTTP successful response code is received.
- For FTP, this is when the client receives a transfer complete.

TotalBytesSent

The total number of bytes sent on the Interface between BOMTime and EOMTime.

TCPOpenRequestTime

The request time in UTC, specified to microsecond precision.

- For HTTP, this is when the TCP socket open (SYN) was sent for the HTTP connection.
- For FTP, this is when the TCP socket open (SYN) was sent for the data connection.

TCPOpenResponseTime

The response time in UTC, specified to microsecond precision.

- For HTTP, this is when the TCP ACK to the socket opening the HTTP connection was received.
- For FTP, this is when the TCP ACK to the socket opening the Data connection was received.

References

For more information about TR-069, see the following external references. Links download the associated PDF or DOC file.

- [\[TR-069 Issue 3\] \(http://www.broadband-forum.org/technical/download/TR-069_Amendment-3.pdf\)](http://www.broadband-forum.org/technical/download/TR-069_Amendment-3.pdf) — TR-069 Amendment 3, CPE WAN Management Protocol
- [\[TR-098 Issue 2\] \(http://www.broadband-forum.org/technical/download/TR-098_Amendment-2.pdf\)](http://www.broadband-forum.org/technical/download/TR-098_Amendment-2.pdf) — TR-098 Internet Gateway Device Data Model for TR-069, Amendment 2
- [\[TR-181\] \(https://www.broadband-forum.org/technical/download/TR-181_Issue-2_Amendment-2.pdf\)](https://www.broadband-forum.org/technical/download/TR-181_Issue-2_Amendment-2.pdf) — TR-181 Device Data Model for TR-069, Amendment 2
- [Wi-Fi Provisioning Framework Specification \(https://www.cablelabs.com/doczone/wireless/requirements/specs/current/wrspwifimgmti02101005.doc\)](https://www.cablelabs.com/doczone/wireless/requirements/specs/current/wrspwifimgmti02101005.doc), WR-SP-WiFi-MGMT-I02-101005

Corporate Headquarters

ARRIS · Suwanee · Georgia · 30024 · USA

T: 1-678-473-2000 F: 1-678-473-8470

www.arris.com