

CVM 2.02 – Call History Report Raw Data Format Explained

Document ID: 13407

Contents

Introduction

Prerequisites

- Requirements
- Components Used
- Conventions

CVM Router File

- Header Record
- Router Record

CVM Call History File

- Header Record
- QoV Trap Record (optional)
- Call History Data Records (optional)
- Telephony Call Leg Data
- VoIP Call Leg Data
- VoATM Call Leg Data
- VoFR Call Leg Data
- Disconnect Causes

Related Information

Introduction

Cisco Voice Manager (CVM) 2.02 uses the TeleMate Quickview reporting package to process the raw data format out of CVM and produce user-readable reports. This document explains the raw data format fields for customers who wish to develop their own reporting tools. It describes the files that are output by CVM 2.02 when it polls routers for Call History Data.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on CVM version 2.02.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

CVM Router File

This file is generated whenever the CVM Poller application starts up or a new router is added to CVM. The CVM Router File has the filename **router.dat**. The file consists of a Header record and at least one Router record. All records consist of fields separated by semicolons.

Header Record

Identifier	CVM Router 2.0
Version	2
Time Stamp	NT Timestamp of when file generated.
Record Count	The number of router records that follow.

Router Record

Router IP Address	IP address of router
Router Name	Name of router, if available.

CVM Call History File

This file is generated for each router within CVM whenever the CVM Poller application receives Quality of Voice (QoV) Traps for a router or extracts Call History information from a router. The CVM Call History File has a filename of the format **r_..dat**. This filename convention ensures a unique file is generated every time CVM pulls Call History data from a router.

The file consists of a Header record and at least one QoV Trap record or Call History Data record. QoV Trap records are only generated when the associated Call History record (as identified by the Connection ID [GUID]) is not in the current Call History Data set. If the Call History record is present, then the QoV Trap information is combined into the VoIP call leg record for the appropriate Connection ID.

All records consist of fields separated by commas.

Header Record

Identifier	CVM2.0
Version	2
Time Stamp	NT Timestamp of poll
Router IP address	IP address of router
Router sysUpTime	Current value of router's sysUpTime in 1/100th seconds. (unsigned long)
QOV record count	The number of QoV Trap records that follow
Record count	The number of Call History records that follow

QoV Trap Record (optional)

Record Type	5
Connection ID (GUID)	hex-string
Icpif	integer

Call History Data Records (optional)

All Call History records have this format:

Record Type	0 - Telephony, 1- VoIP, 2 - VoATM,
3 - VoFR	

Source IP address (if from phone)	IP address of source router or null
Destination IP address null (if to phone)	IP address of destination router or null
Connection ID (GUID)(cvCallHistoryConnectionId)	Hex-string
Calling/Called Number (cCallHistoryPeerAddress)	Digit-string
Sub-address (cCallHistoryPeerSubAddress)	String
Disconnect Text(cCallHistoryDisconnectText) (see list at end of document)	Textual value of disconnect cause
Connect Time (cCallHistoryConnectTime)	Call connect time mm/dd/yy hh:mm:ss
Disconnect Time (cCallHistoryDisconnectTime) hh:mm:ss	Call disconnect time mm/dd/yy
Origin (cCallHistoryCallOrigin) 3 - callback	1 - originate, 2 - answer,
Charged Units (cCallHistoryChargedUnits)	Unsigned long
Call Type (cCallHistoryInfoType)	2 - voice, 10 - fax
Transmit Packets (cCallHistoryTrasmitPackets)	Unsigned long
Transmit Bytes (cCallHistoryTransmitBytes)	Unsigned long
Receive Packets (cCallHistoryReceivePackets)	Unsigned long
Receive Bytes (cCallHistoryReceiveBytes)	Unsigned long
Call Leg Specific Data	See following definitions.

Telephony Call Leg Data

Transmit Duration (cvCallHistoryTxDuration)	Unsigned long
Speech Duration (cvCallHistoryVoiceTxDuration)	Unsigned long
Fax Duration (cvCallHistoryFaxTxDuration)	Unsigned long
Coder Rate (cvCallHistoryCoderTypeRate)	Coder rate used during call:
1 - other	
2 - fax2400	
3 - fax4800	
4 - fax7200	
5 - fax9600	
6 - fax14400	
7 - fax12000	
10 - g729r8000	
11 - g729Ar8000	
12 - g726r16000	
13 - g726r24000	
14 - g726r32000	
15 - g711ulawr64000	
16 - g711Alawr64000	
17 - g728r16000	
18 - g723r6300	
19 - g723r5300	
20 - gsmr13200	
21 - g729Br8000	
22 - g729ABr8000	
23 - g723Ar6300	
24 - g723Ar5300	
25 - ietfg729r8000	
Noise Level (cvCallHistoryNoiseLevel)	Integer
ACOM Level (cvCallHistoryACOMLevel)	Integer
Session Target (cvCallHistorySessionTarget)	String
Image Page Count (cvCallHistoryImgPageCount)	Unsigned long
PeerId (cCallHistoryPeerId)	Integer
PeerIfIndex (cCallHistoryPeerIfIndex)	Integer
LogicalIfIndex (cCallHistoryLogicalIfIndex)	Integer
PeerIfName	String
LogicalIfName	String
Call Setup Time (cCallHistorySetupTime)	Call setup time mm/dd/yy hh:mm:ss

VoIP Call Leg Data

Vad Enabled (cvCommonDcCallHistoryVADEnabled)	1 - true, 2 - false
Coder Rate (cvCommonDcCallHistoryCoderTypeRate)	See Telephony Call Leg Data
Codec Bytes (cvCommonDcCallHistoryCodecBytes)	Integer
InBand Signaling (cvCommonDcCallHistoryInbandSignalling)	1 - cas, 2 - external, 3 - cept,
4 - transparent	
Remote IP address (cvVoIPCallHistoryRemoteIPAddress)	IP address
Remote UDP port (cvVoIPCallHistoryRemoteUDPPort)	Integer
Round Trip Delay (cvVoIPCallHistoryRoundTripDelay)	Unsigned long
Selected Qos (cvVoIPCallHistorySelectedQos)	1 - best effort,
2 - controlled-load, 3- guaranteed-load	
Protocol (cvVoIPCallHistorySessionProtocol)	1 - other, 2 - cisco,
3- sdp, 4-sip	
Session Target (cvVoIPCallHistorySessionTarget)	String
Icpif (cvVoIPCallHistoryIcpif)	Integer
Lost Packets (cvVoIPCallHistoryLostPackets)	Unsigned long
Late Packets (cvVoIPCallHistoryLatePackets)	Unsigned long
Early Packets (cvVoIPCallHistoryEarlyPackets)	Unsigned long
QOVTrap	0 - no, 1 - yes
QOVTrap Icpif	Integer
OnTimePlayOut (cvVoIPCallHistoryOnTimRvPayout)	Unsigned long
GapFilledWithSilence (cvVoIPCallHistoryGapFilledWithSilence)	Unsigned long
GapFilledWithPrediction (cvVoIPCallHistoryGapFilledWithPrediction)	Unsigned long
GapFilledWithInterpolation (cvVoIPCallHistoryGapFilledWithInterpolation)	Unsigned long
GapFilledWithRedundancy (cvVoIPCallHistoryGapFilledWithRedundancy)	Unsigned long
HiWaterDelay (cvVoIPCallHistoryHiWaterPayoutDelay)	Unsigned long
LoWaterDelay (cvVoIPCallHistoryLoWaterPayoutDelay)	Unsigned long
ReceiveDelay (cvVoIPCallHistoryReceiveDelay)	Unsigned long
PeerId (cCallHistoryPeerID)	Integer
PeerIfIndex (cCallHistoryPeerIfIndex)	Integer
LogicalIfIndex (cCallHistoryLogicalIfIndex)	Integer
PeerIfName	String
LogicalIfName	String
Call Setup Time (cCallHistorySetupTime)	Call setup time mm/dd/yy
hh:mm:ss	

VoATM Call Leg Data

Vad Enabled (cvCommonDcCallHistoryVADEnabled)	1 - true, 2 - false
Coder Rate (cvCommonDcCallHistoryCoderTypeRate)	See Telephony Call Leg Data
Codec Bytes (cvCommonDcCallHistoryCodecBytes)	Integer
InBand Signaling (cvCommonDcCallHistoryInbandSignaling)	1 - cas, 2 - external, 3 - cept,
4 - transparent	
Interface Name (cvAtmCallHistoryInterfaceName)	String
Session Target (cvAtmCallHistorySessionTarget)	String
Vpi (cvAtmCallHistoryVpi)	Integer
Vci (cvAtmCallHistoryVci)	Integer
Sub-channel id (cvAtmCallHistoryCid)	Integer
Protocol (cvAtmCallHistorySessionProtocol)	1 - cisco-switched, 2 - frfl1
DTMF Relay (cvAtmCallHistoryDtmfRelay)	1 - true, 2 - false
Sequence Numbers (cvAtmCallHistorySequenceNumbers)	1 - true, 2 - false
PeerId (cCallHistoryPeerID)	Integer
PeerIfIndex (cCallHistoryPeerIfIndex)	Integer
PeerIfName	String
LogicalIfName	String
Call Setup Time (cCallHistorySetupTime)	Call setup time mm/dd/yy
hh:mm:ss	

VoFR Call Leg Data

Vad Enabled (cvCommonDcCallHistoryVADEnabled)	1 - true, 2 - false
Coder Rate (cvCommonDcCallHistoryCoderTypeRate)	See Telephony Call Leg Data
Codec Bytes (cvCommonDcCallHistoryCodecBytes)	Integer
InBand Signaling (cvCommonDcCallHistoryInBandSignaling)	1 - cas, 2 - external, 3 - cept, 4 - transparent
Interface Name (cvFrCallHistoryInterfaceName)	String
Session Target (cvFrCallHistorySessionTarget)	String
Dlci (cvFrCallHistoryDlci)	Integer
Sub-channel id (cvFrCallHistoryCid)	Integer
Protocol (cvFrCallHistorySessionProtocol)	1 - cisco-switched, 2 - frfl1
DTMF Relay (cvAtmCallHistoryDtmfRelay)	1 - true, 2 - false
Sequence Numbers (cvFrCallHistorySequenceNumbers)	1 - true, 2 - false
PeerId (cCallHistoryPeerID)	Integer
PeerIfIndex (cCallHistoryPeerIfIndex)	Integer
LogicalIfIndex (cCallHistoryLogicalIfIndex)	Integer
PeerIfName	String
LogicalIfName	String
Call Setup Time (cCallHistorySetupTime)	Call setup time mm/dd/yy hh:mm:ss

Disconnect Causes

This list describes the disconnect text strings (as currently defined in the Cisco IOS Software code) that are output in the Disconnect Text field of each Call History Data record:

```
char *cause_strings[] = {
    "", /* 0 */
    "unassigned number.", /* 1 */
    "no_route_to_transit_network", /* 2 */
    "no route to destination.", /* 3 */
    "send info tone", /* 4 */
    "misdialled trunk prefix", /* 5 */
    "channel unacceptable", /* 6 */
    "call awarded", /* 7 */
    "preemption", /* 8 */
    "preemption reserved", /* 9 */
    "n/a", /* 10 */
    "n/a", /* 11 */
    "n/a", /* 12 */
    "n/a", /* 13 */
    "n/a", /* 14 */
    "n/a", /* 15 */
    "normal call clearing.", /* 16 */
    "user busy.", /* 17 */
    "no user response.", /* 18 */
    "no user answer.", /* 19 */
    "subscriber absent", /* 20 */
    "call rejected.", /* 21 */
    "number changed", /* 22 */
    "n/a", /* 23 */
    "n/a", /* 24 */
    "n/a", /* 25 */
    "non selected user clearing", /* 26 */
    "destination out of order", /* 27 */
    "invalid number.", /* 28 */
    "facility rejected", /* 29 */
    "response to status enquiry", /* 30 */
    "normal, unspecified.", /* 31 */
    "n/a", /* 32 */
    "n/a", /* 33 */
    "no circuit.", /* 34 */
    "requested vpci vci not available", /* 35 */
}
```

"vpci vci assignment failure",	/* 36 */
"cell rate not available",	/* 37 */
"network out of order",	/* 38 */
"perm frame mode out of service",	/* 39 */
"perm frame mode operational",	/* 40 */
"temporary failure",	/* 41 */
"switch congestion",	/* 42 */
"access info discarded",	/* 43 */
"no requested circuit.",	/* 44 */
"no vpci vci available",	/* 45 */
"precedence call blocked",	/* 46 */
"no resource.",	/* 47 */
"n/a",	/* 48 */
"qos unavailable",	/* 49 */
"facility not subscribed",	/* 50 */
"n/a",	/* 51 */
"n/a",	/* 52 */
"cug outgoing calls barred",	/* 53 */
"n/a",	/* 54 */
"cug incoming calls barred",	/* 55 */
"n/a",	/* 56 */
"bearer capability not authorized",	/* 57 */
"bearer capability not available",	/* 58 */
"n/a",	/* 59 */
"n/a",	/* 60 */
"n/a",	/* 61 */
"inconsistency in info and class",	/* 62 */
"service or option not available, unspecified",	/* 63 */
"n/a",	/* 64 */
"bearer capability not implemented",	/* 65 */
"chan type not implemented",	/* 66 */
"n/a",	/* 67 */
"n/a",	/* 68 */
"facility not implemented",	/* 69 */
"restricted digital info bc only",	/* 70 */
"n/a",	/* 71 */
"n/a",	/* 72 */
"n/a",	/* 73 */
"n/a",	/* 74 */
"n/a",	/* 75 */
"n/a",	/* 76 */
"n/a",	/* 77 */
"n/a",	/* 78 */
"service not implemented",	/* 79 */
"n/a",	/* 80 */
"invalid call ref value",	/* 81 */
"channel does not exist",	/* 82 */
"call exists call id in use",	/* 83 */
"call id in use",	/* 84 */
"no call suspended",	/* 85 */
"call cleared",	/* 86 */
"user not in cug",	/* 87 */
"incompatible destination",	/* 88 */
"n/a",	/* 89 */
"non existent cug",	/* 90 */
"invalid transit network",	/* 91 */
"n/a",	/* 92 */
"aal parms not supported",	/* 93 */
"n/a",	/* 94 */
"invalid message",	/* 95 */
"mandatory ie missing",	/* 96 */
"message type not implemented",	/* 97 */
"message type not compatible",	/* 98 */
"ie not implemented",	/* 99 */
"invalid ie contents",	/* 100 */
"message in incomp call state",	/* 101 */

```
"recovery on timer expiry",          /* 102 */
"non implemented param passed on",   /* 103 */
"n/a",                               /* 104 */
"n/a",                               /* 105 */
"n/a",                               /* 106 */
"n/a",                               /* 107 */
"n/a",                               /* 108 */
"n/a",                               /* 109 */
"unrecognized param msg discarded",  /* 110 */
"protocol error",                   /* 111 */
};
```

Related Information

- **CiscoWorks Voice Manager**
- **Technical Support & Documentation – Cisco Systems**

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2010 – 2011 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Jan 08, 2007

Document ID: 13407
