Avaya Aura® Communication Manager SNMP Renewal Quick Reference Guide

1.0 Introduction

This White Paper outlines the changes made to the CM SNMP Stack, CM SNMP Subagents, CM SNMP Commands, and CM SNMP System Management Interface (SMI) Pages in CM Release 6.3.1xx and CM Release 7.0. The G3-MIB was retired in CM Release 6.3.1x. Two new MIBs, the AVAYA-AURA-CM-MIB and the AVAYA-AURA-CMALARM-MIB replace it. In addition, the SNMP stack was changed to Net-SNMP and new Net-SNMP CM Subagents were developed. This document does not include basic SNMP information. It is expected that users already understand basic SNMP.

2.0 General Information

2.1 snmpd (MasterAgent)

The new CM Master Agent is from Net-SNMP. As with the previous implementation the snmpd is configured to listen on port 161 and supports both IPv4 and IPv6 queries and outgoing trap and inform notification messages. The snmpd uses the snmpd.conf configuration file (see section 2.5). More information about the Net-SNMP snmpd can be found at:

http://www.net-snmp.org/docs/man/snmpd.html

2.2 MIB2-SubAgent

The MIB2 SubAgent is provided by Net-SNMP and is embedded in the Net-SNMP snmpd (Master Agent). The ipAddressTable and the ipNetToPhysicalTable are disabled on CM due to performance issues.

2.3 snmptrapd (SNMPManager)

CM continues to use the Net-SNMP snmptrapd as its incoming trap receiver, also known on CM as the SNMPManager. However, in past releases of CM, incoming trap authorization was disabled. Therefore, CM could not support SNMPv3 traps and would accept any SNMPv1/v2 traps that were sent to it. To make CM more secure, support of

authorization checking on incoming SNMP traps was added beginning in CM Release 6.3.1xx. This feature is disabled by default. However, it can be enabled using the *snmpinctrapconfig –enable* or disabled using the *snmpinctrapconfig –disable* command. Please see the *snmpinctrapconfig* command (section X.X) for more details.
As in the past, the snmptrapd is configured to listen on port 162 and supports both IPv4 and IPv6 incoming trap and inform notification messages. The snmptrapd uses the snmptrapd.conf configuration file (see section 2.6). More information about the Net-SNMP snmptrapd can be found at:

http://www.net-snmp.org/docs/man/snmptrapd.html

2.4 Standard MIBs

Currently CM Release 6.3.1xx is running Net-SNMP Release 5.3.2.2 and CM Release 7.0 is Running Net-SNMP Release 5.5. The following link details what standard MIBS each release of Net-SNMP supports:

http://www.net-snmp.org/docs/README.agent-mibs.html

2.5 snmpd.conf File

The snmpd.conf file is the configuration file used by the Net-SNMP snmpd (Master Agent). CM installs a default snmpd.conf file during initial installation. The snmpd.conf file is root protected. All SNMP access and outgoing trap administration is configured using either the CM SNMP ***snmpuserconfig*** and ***snmptrapconfig*** commands or the CM SNMP ***Access*** and CM SNMP ***FP Traps*** SMI Pages.

The following directives are configured in the snmpd.conf file.

authtrapenable is set to enabled.
maxGetbulkRepeats is set to 0
maxGetbulkResponses is set to 0

2.6 snmptrapd.conf File

The snmptrapd.conf file is the configuration file used by the snmptrapd (SNMPManager). CM installs a default snmptrapd.conf file during initial installation. The snmptrapd.conf file is root protected. All SNMP incoming trap administration is configured using the either the CM SNMP ***snmpinctrapconfig*** command or the CM SNMP ***Incoming Traps*** SMI Page. Incoming trap authorization is disabled by default.

2.7 SystemDescription and SystemOID

Two new SysDescr and SysObjectIDs were added in CM Release 6.3.1xx:

1. SystemDescription *Avaya Aura® Communication Manager SP* associated with SystemOID *avayaAuraSp* (*1.3.6.1.4.1.6889.1.73.1*), was added to support CM Servers using System Platform.
2. SystemDescription *Avaya Aura ® Communication Manager VE* associated with SystemOID *avayaAuraVe* (*1.3.6.1.4.1.6889.1.73.2*), was added to support CM Servers using virtualization.

2.8 Upgrades and Migrations

No upgrade or migration path of SNMP configuration data is provided from previous releases of CM. Therefore users will have to re-administer SNMP configuration and filter information. As in the previous implementation, SNMP administration data on duplicated systems are not file synced. Therefore both servers must be administered and should contain the same administration information.

2.9 Net-SNMP Commands

Net-SNMP commands such as snmpwalk or snmpset should never be used locally on a CM Server. In addition, the Net-SNMP snmpconf command is not to be used to configure SNMP.

3.0 CMSubAgent

The MVSubAgent was replaced by the CMSubAgent. The CMSubAgent is responsible for retrieving product specific data from the CM Server. The CMSubAgent was redesigned to support internal caching of CM data. Internal caching was added to improve performance. Internal caching limits the frequency of queries from the CMSubAgent to CM. The caching intervals are fixed and cannot be modified. Each MIB Group's Caching interval is outlined in the AVAYA-AURA-CM-MIB.

As previously stated the G3-MIB was retired and replaced by the AVAYA-AURA-CM-MIB. To increase system performance the AVAYA-AURA-CM-MIB was designed with the following enhancements:

1. The payload size, on some MIB Groups, was increased to reduce the number of queries needed to retrieve SNMP data. The avCmConfiguration MIB Group is a good example of this. Instead of supporting up to 64 individual port OIDs per board, the avCmConfiguration MIB Group concatenates all 64 ports into one OID and returns all of the port data in a single string.
2. Six MIB Groups, the avCmListStationRange, avCmListTrkRange, avCmListMemTrunkRange, avCmStatusTrunkRange, avCmStatusStationRange, and, avCmListIpUnregisteredRange, require that range information be inputted in the form of a starting location and a count, in order to query them. The starting location is the record you want to start the query at. The count is the number of records you want to retrieve. Range restricted MIB Groups were added to limit the amount of data that is requested from CM in a single query. Each range

restricted MIB Group has a limit on the maximum number of records that can be retrieved in a single query. See the avCmXxxXxxxRangeCount object for each MIB Group in AVAYA-AURA-CM-MIB for more details.

3. Some MIB Groups are now limited to the amount of data they return. To increase performance and to limit the impact querying certain MIB Groups have on system resources, limitations on the amount of data returned by the avCmListStation, avCmListMemTrunk, avCmStatusTrunk, avCmStatusStation, and avCmListIpUnregistered MIB groups are restricted to 500 records. Users should use range restricted MIB Groups if they need to query more than 500 records.

Another very important change is that the documentation for each MIB Group is embedded as comments in the MIB. The following example shows the documentation for the avCmVersion MIB Group (in green) embedded in the AVAYA-AURA-CM-MIB:

```
-- ***************************    avCmVersion Group ************************

-- MIB GROUP DESCRIPTION

-- MIB Group Name:
--              avCmVersion
-- OID:
--          enterprises.6889.2.73.8.1.1
-- Description:
--          The avCmVersion MIB Group displays the output of the “swversion”
--          command. The “swversion” command displays the Servers Software
--          Version information
-- Releases Supported:
--          CM6.3.100 and greater
-- Polling Type:
--          Configuration
-- Polling Interval:
--          No more than once a day.
-- Default Caching Interval
--          60 seconds
-- Timeout Interval:
--          Initial Timeout should be set to 15 seconds.
-- Retry Interval:
--          Initial Retry should be set to 3.
-- Limitations:
--          Green/Low – The data collected by this MIB Group
--          is minimal and therefore has a low impact on the system.
-- Command Type:
--          1. BASH
-- Command:
--          1. swversion
-- Form/Report Name:
```



```
--            1. CM Software Version
-- Reference Document:
--            1. A detailed description of the command and form can be
--               found in document referenced by REF_DOC_1 in the
--               DOCUMENT REFERENCE SECTION.
-- Command Permission/User-profile:
--            1. No special permissions needed.

  avCmVersion           OBJECT IDENTIFIER ::= { avCmObjects 1 }

avCmVersionOperSystem   OBJECT-TYPE
   SYNTAX      SnmpAdminString (SIZE (0..50))
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
      "
      Fld OID Name:   avCmVersionOperSystem
      Fld OID:        enterprises.6889.2.73.8.1.1.1
      Fld Form/Rpt:   CM Software Version
      Fld Format:     Single Field
      Fld(s) Name:    Operating System
      Fld Descr:      An object that contains the operating system
                      running on the Server.
      "
   ::= {    avCmVersion 1 }

  :
  :
```

Users should reference the above documentation before attempting to query data from CM via SNMP. Copies of the AVAYA-AURA-CM-MIB and the AVAYA-AURA-CMALARM-MIB files are loaded on the CM Server. They are located in the directory /usr/share/snmp/mibs or they can be viewed or downloaded from the SNMP *Access* and SNMP *FP Traps* SMI Pages. They can also be downloaded from https://support.avaya.com.

Note: The entire AVAYA-AURA-CM-MIB should never be walked in one snmpwalk command. Users should only walk individual MIB Groups, specific rows, or leaf nodes.

The following table outlines the changes in the MIB Groups (OIDs) from the G3-MIB to the new AVAYA-AURA-CM-MIB. Many changes have been made. OIDs highlighted in red have been deprecated. OIDs highlighted in blue are not supported in the AVAYA-AURA-CM_MIB. OIDs highlighted in green are new MIB Groups added to the AVAYA-AURA-CM-MIB. OIDs that are not highlighted are OIDs that are supported in both the G3-MIB and the AVAYA-AURA-CM-MIB.

G3-MIB 1.3.6.1.4.1.6889.2.8.1		AVAYA-AURA-CM-MIB 1.3.6.1.4.1.6889.2.73.1	
OID	**MIB Group Name**	**OID**	**MIB Group Name**
1	**g3-connect**	**obsolete**	
2	**g3vintage**	**obsolete**	**Deprecated some leafs reassigned to new MIB Groups**
2.5	g3vintageSpeArelease	obsolete	
2.6	g3vintageSpeBrelease	obsolete	
2.7	g3vintageCurMemory	1.9	avCmVersionMemConfig
2.8	g3vintageSpeAupID	obsolete	
2.9	g3vintageSpeBupID	obsolete	
2.10	g3vintageSpeAupState1	obsolete	
2.11	g3vintageSpeBupState1	obsolete	
2.12	g3vintageSpeAupState2	obsolete	
2.13	g3vintageSpeBupState2	obsolete	
2.14	g3vintageVSPacketBus	obsolete	
2.15	g3vintageOfferCategory	obsolete	
2.16	g3vintageATMPnc	obsolete	
2.17	g3vintageProductID	4.1	avCmServerInfoProductID
2.18	g3vintageOSSNumber1	obsolete	
2.19	g3vintageAbbAlmRep1	obsolete	
2.20	g3vintageOSSNumber2	obsolete	
2.21	g3vintageAbbAlmRep2	obsolete	
2.22	g3vintageAOActivate	obsolete	
2.23	g3vintageCANotify	obsolete	
2.24	g3vintageResNotify	obsolete	
2.25	g3vintagePbxID	obsolete	
2.26	g3vintageACAFlag	NS	
2.27	g3vintageIPAddress1	obsolete	
2.28	g3vintageIPAddress2	obsolete	
2.29	g3vintageIPAddress3	obsolete	
2.30	g3vintageIPAddress4	obsolete	avCmSysParamCustPortNetSup
2.31	g3vintagePortNetSupport	103.5	avCmSysParamCustLSP
2.32	g3vintageLocalSpareProc	103.3	avCmSysParamCustPlat
2.33	g3vintagePlatform	103.1	avCmCapacityUsageIPStatUsed
2.34	g3vintageLocalNodeNbr	obsolete	avCmCapacityUsageIPStatAvail
2.35	g3vintageIPStaReg	21.1	avCmCapacityUsageIPStatSysLim
2.36	g3vintageIPStaAvail	21.2	avCmCapacityUsageIPAtConUsed
2.37	g3vintageIPStaLimit	21.3	avCmCapacityUsageIPAtConAvail
2.38	g3vintageIPAttdReg	21.4	avCmCapacityUsageIPAtConSysLim
2.39	g3vintageIPAttdAvail	21.5	avCmCapacityUsageRemOffStatUsed

2.40 **2.41** **2.42** **2.43**	g3vintageIPAttdLimit g3vintageRemoteOffReg g3vintageRemoteOffAvail g3vintage RemoteOffLimit	21.6 21.7 21.8 21.9	avCmCapacityUsageRemOffStatAvail avCmCapacityUsageRemOffStatSysLim
3	**g3config**	**3**	**avCmConfiguration**
4	**g3alarms**	**5**	**avCmAlarms**
5	**g3errors**	**6**	**avCmErrors**
6	**g3health**	**29**	**avCmStatusHealth**
7	**g3cabinet**	**NS**	
new		**4**	**avCmServerInfo**
8	**g3cabtype**	**obsolete**	
new		**8**	**avCmListStationRange**
9	**g3cartype**	**obsolete**	
10	**g3port**	**obsolete**	
11	**g3station**	**7**	**avCmListStation**
new		**11**	**avCmListTrunkRange**
12	**g3statsta**	**30**	**avCmStatusStation**
new		**12**	**avCmCapacityAsai**
13	**g3trunkmem**	**24**	**avCmListMemTrunk**
new		**13**	**avCmCapacityBcms**
14	**g3trunksta**	**26**	**avCmStatusTrunk**
new		**14**	**avCmCapacityCallVec**
15	**g3datamod**	**obsolete**	
new		**15**	**avCmCapacityDialPlan**
16	**g3datamsta**	**obsolete**	
new		**16**	**avCmCapacityHuntGrps**
new	**OID 17 is not in G3-MIB**	**17**	**avCmCapacityAnnounce**
18	**g3timedate**	**56**	**avCmDispTime**
new		**18**	**avCmCapacityTrunksGroup**
19	**g3busytrk**	**obsolete**	
/new		**19**	**avCmCapacityVoiceTerm**
20	**g3busybrd**	**obsolete**	
new		**20**	**avCmCapacityLicGroup**
21	**g3servalm**	**2**	**avCmServerAlarm**
new		**21**	**avCmCapacityIPUsage**
22	**g3msgalm**	**57**	**avCmMessagingAlarm**
new		**22**	**avCmCapacityIPUsage**

23	g3ipevt	NS	
new		23	avCmCapacityCurrentUsage
24	g3platcmds	83	avCmPlatCmds
25	g3version	1	avCmVersion
new		25	avCmListMemTrunkRange
26	g3update	9	avCmUpdate
27	g3partition	obsolete	
New	OID 28 is not in G3-MIB	28	avCmStatusTrunkMem
None	OID 29 is not in G3-MIB		
30	g3ds1cfg	NS	
31	g3extdev	NS	
32	g3trunkgrp	10	avCmListTrunk
33	g3bulletin	obsolete	
new		33	avCmMeasOccLstHr
34	g3ds1	NS	
new		34	avCmMeasOccBusIntvl
35	g3atmpnc	obsolete	
36	g3pnchealth	NS	
new		36	avCmMeasTrkGrpSumToday
new		84	avCmListSigGrp
37	g3siggroup	85	avCmStatusSigGrp
38	g3restart	58	avCmInitCauses
39	g3fiblerlink	obsolete	
new		39	avCmMeasOutTrkToday
40	g3routepattern	NS	
41	g3trunkcfg	NS	
42	g3stationcfg	NS	
new		42	avCmMeasLightUseTrkToday
43	g3atmcfg	obsolete	
44	g3aca	NS	
45	g3atmtrunk	obsolete	
46	g3atmportpg1	obsolete	
47	g3atmportpg2	obsolete	
48	g3atmportpg2	obsolete	
49	g3dmodule	obsolete	
50	g3occsum	32	avCmMeasOccSum
51	g3attgrp	NS	
52	g3attpos	NS	
53	g3trunksum	35	avCmMeasTrkGrpSumLstHr
54	g3pktrunksum	37	avCmMeasTrkGrpSumYest
new		54	avCmListAnn
55	g3trkwbsum	obsolete	

new		55	avCmListIpUnregisteredRange
56	g3pktrunkwbsum	obsolete	
57	g3pktrunkout	40	avCmMeasOutTrkYest
58	g3ptrunklight	43	avCmMeasLightUsedYest
59	g3pnblock	NS	
60	g3pkpnblock	NS	
new		60	avCmMeasAnnIntegAllToday
61	g3snblock(Obsolete)	None	
62	g3pksnblock(Obsolete)	None	
		63	avCmListIntegAnn
63	g3hunt	87	avCmListMeasHuntGrpLstHr
new		88	avCmListMeasHuntGrpToday
64	g3pkhunt	89	avCmListMeasHuntGrpYest
65	g3huntlist	86	avCmListHuntGrp
66	g3tonerec	97	avCmListMeasToneRecLstHr
new		98	avCmListMeasToneRecToday
67	g3pktonerec	99	avCmListMeasToneRecYest
new		67	avCmMeasIpVoStatHrData
68	g3tonerecsum	100	avCmListMeasToneRecSumLstHr
new		101	avCmListMeasToneRecSumToday
new		102	avCmListMeasToneRecSumYest
new		68	avCmMeasIpVoStatSumJitterLastHr
69	g3pktonerecsum	NS	
new		69	avCmMeasIpVoStatSumJitterToday
70	g3rpatcfg	NS	
new		70	avCmMeasIpVoStatSumJitterYest
71	g3rpat	NS	
new		71	avCmMeasIpVoStatSumPkLosLstHr
72	g3pkrpat	NS	
new		72	avCmMeasIpVoStatSumPkLosToday
73	g3secviolate	44	avCmMeasSecViolationSum
new		73	avCmMeasIpVoStatSumPkLosYest
74	g3cbctrunk	NS	
new		74	avCmMeasIpVoStatSumRtdelLstHr
75	g3deftime	obsolete	
new		75	avCmMeasIpVoStatSumRtdelToday
76	g3trunkout	38	avCmMeasOutTrkLstHr
new		76	avCmMeasIpVoStatSumRtdelYest
77	g3trunklight	41	avCmMeasLightUseTrkLstHr
new		77	avCmMeasIpVoStatSumDataLstHr
78	g3loadtotal	NS	

new		**78**	**avCmMeasIpVoStatSumDataToday**
79	**g3loadint**	**NS**	
new		**79**	**avCmMeasIpVoStatSumDataYest**
80	**g3pkloadinc**	**NS**	
81	**g3loadout**	**NS**	
82	**g3loadtan**	**NS**	
83	**g3pkloadtotal**	**NS**	
84	**g3pkloadint**	**NS**	
85	**g3pkloadinc**	**NS**	
86	**g3pkloadout**	**NS**	
87	**g3pkloadtan**	**NS**	
88	**g3atmlatency**	**obsolete**	
89	**g3pkatmlatency**	**obsolete**	
90	**g3ipcodecsum**	**NS**	
91	**g3pkipcodecsum**	**NS**	
92	**g3ipregion**	**95**	**avCmListIPNetRegMonitor**
93	**g3ipdspsum**	**NS**	
94	**g3pkipdssum**	**NS**	
95	**g3ipsignal**	**NS**	
96	**g3pkipsignal**	**NS**	
97	**g3annall**	**80**	
new		**81**	**avCmMeasAnnAllToday**
98	**g3pkannall**	**82**	**avCmMeasAnnAllYest**
99	**g3anninteg**	**59**	**avCmMeasAnnIntegAllLstHr**
100	**g3pkanninteg**	**61**	**avCmMeasAnnIntegAllYest**
101	**g3nodename**	**51**	**avCmListNodeNamesAll**
102	**g3trkgrpmem (duplicate)**	**obsolete**	
new		**103.2**	**avCmSysParamCustPlat**
new		**103.4**	**avCmSysParamCustTenantPart**
103	**g3ipinterface**	**90**	
104	**g3gateway**	**45**	**avCmListMedGateway**
105	**g3mediacfg**	**46**	**avCmListConfigMedGateway**
106	**g3stamedia**	**47**	**avCmStatusMedGateway**
107	**g3stamgann**	**62**	**avCmStatusMGAnn**
108	**g3ipserver**	**93**	**avCmListIpServerIntf**
109	**g3regipstat**	**obsolete**	
110	**g3callratedata**	**NS**	
111	**g3callratevoice**	**NS**	
112	**g3callratsrv**	**NS**	
113	**g3callratemedia**	**NS**	
114	**g3callratetotal**	**NS**	
115	**g3clansocsum**	**NS**	

116	**g3pkclansocsum**	**NS**	
117	**g3clanether**	**NS**	
118	**g3clanppp**	**NS**	
119	**g3mmisum**	**NS**	
120	**g3pkmmisum**	**NS**	
121	**g3esmsum**	**NS**	
122	**g3pkesmsum**	**NS**	
123	**g3voicesum**	**NS**	
124	**g3pkvoicesum**	**NS**	
125	**g3ipintlist**		**Obsolete (Duplicate of 103)**
new		**91**	**avCmListInterfaceMedpro**
126	**g3stamedlist (duplicate)**	**obsolete**	
127	**g3lsplist (duplicate)**	**obsolete**	
128	**g3eventhour**	**NS**	
129	**g3eventday**	**NS**	
130	**g3statregion**	**96**	**avCmStatIpNetRegion**
131	**g3ipregion**	**NS**	
132	**g3statmedpro**	**92**	**avCmStatusMedproBrd**
133	**g3ipunreg**	**48**	**avCmListIpUnregistered**
134	**g3commlink**	**NS**	
135	**g3statlink**	**94**	**avCmStatusLink**
136	**g3covpath**	**NS**	
137	**g3pkcovpath**	**NS**	
138	**g3principal**	**NS**	
139	**g3pkprincipal**	**NS**	
140	**g3esmain**	**obsolete**	
141	**g3esservers**	**obsolete**	
142	**g3esparms**	**obsolete**	
143	**g3mgrecrule**	**obsolete**	
144	**g3commproc**	**obsolete**	
145	**g3intvl**	**obsolete**	
146	**g3jitter**	**64**	**avCmMeasIpVoStatHrJitter**
147	**g3pkloss**	**65**	**avCmMeasIPVoStatHrPkLos**
148	**g3rtdelay**	**66**	**avCmMeasIpVoStatHrRtDel**
149	**g3esmainsrv**	**obsolete**	
150	**g3survprocess**	**50**	**avCmDispSurvProcess**
151	**g3nodenameipv4**	**52**	**avCmListNodeNamesV4**
152	**g3nodenameipv6**	**53**	**avCmListNodeNamesV6**
153	**g3lspall**	**49**	**avCmListSurvProcess**
154	**g3trunkstatOptions**	**27**	**avCmStatusTrunkRange**
155	**g3statstaOptions**	**31**	**avCmStatusStationRange**

250	**g3busyrls**	**obsolete**	

Figure 1 – G3-MIB OID to AVAYA-AURA-CM-MIB OID Conversion Table

The table below outlines the G3-MIB Groups that were deprecated

G3-MIB **1.3.6.1.4.1.6889.2.8.1**		**AVAYA-AURA-CM-MIB** **1.3.6.1.4.1.6889.2.73.1**	
OID	**MIB Group Name**	**OID**	**MIB Group Name**
1	**g3-connect**	**obsolete**	
2	**g3vintage**	**obsolete**	**Deprecated some leafs reassigned to new MIB Groups**
2.5	g3vintageSpeArelease	obsolete	
2.6	g3vintageSpeBrelease	obsolete	
2.8	g3vintageSpeAupID	obsolete	
2.9	g3vintageSpeBupID	obsolete	
2.10	g3vintageSpeAupState1	obsolete	
2.11	g3vintageSpeBupState1	obsolete	
2.12	g3vintageSpeAupState2	obsolete	
2.13	g3vintageSpeBupState2	obsolete	
2.14	g3vintageVSPacketBus	obsolete	
2.15	g3vintageOfferCategory	obsolete	
2.16	g3vintageATMPnc	obsolete	
2.18	g3vintageOSSNumber1	obsolete	
2.19	g3vintageAbbAlmRep1	obsolete	
2.20	g3vintageOSSNumber2	obsolete	
2.21	g3vintageAbbAlmRep2	obsolete	
2.22	g3vintageAOActivate	obsolete	
2.23	g3vintageCANotify	obsolete	
2.24	g3vintageResNotify	obsolete	
2.25	g3vintagePbxID	obsolete	
2.27	g3vintageIPAddress1	obsolete	
2.28	g3vintageIPAddress2	obsolete	
2.29	g3vintageIPAddress3	obsolete	
2.30	g3vintageIPAddress4	obsolete	
2.34	g3vintageLocalNodeNbr	obsolete	
8	**g3cabtype**	**obsolete**	
9	**g3cartype**	**obsolete**	
10	**g3port**	**obsolete**	
15	**g3datamod**	**obsolete**	
16	**g3datamsta**	**obsolete**	

19	g3busytrk	obsolete	
20	g3busybrd	obsolete	
27	g3partition	obsolete	
33	g3bulletin	obsolete	
35	g3atmpnc	obsolete	
39	g3fiblerlink	obsolete	
43	g3atmcfg	obsolete	
45	g3atmtrunk	obsolete	
46	g3atmportpg1	obsolete	
47	g3atmportpg2	obsolete	
48	g3atmportpg2	obsolete	
49	g3dmodule	obsolete	
55	g3trkwbsum	obsolete	
56	g3pktrunkwbsum	obsolete	
61	g3snblock(Obsolete)	None	
62	g3pksnblock(Obsolete)	None	
75	g3deftime	obsolete	
88	g3atmlatency	obsolete	
89	g3pkatmlatency	obsolete	
102	g3trkgrpmem (duplicate)	obsolete	
109	g3regipstat	obsolete	
126	g3stamedlist (duplicate)	obsolete	
127	g3lsplist (duplicate)	obsolete	
140	g3esmain	obsolete	
141	g3esservers	obsolete	
142	g3esparms	obsolete	
143	g3mgrecrule	obsolete	
144	g3commproc	obsolete	
145	g3intvl	obsolete	
149	g3esmainsrv	obsolete	
250	g3busyrls	obsolete	

Figure 2 – G3-MIB OID that were deprecated

The following table details the mapping of the AVAYA-AURA-CM-MIB OIDs to G3-MIB OIDs:

AVAYA-AURA-CM-MIB 1.3.6.1.4.1.6889.2.73.1		G3-MIB 1.3.6.1.4.1.6889.2.8.1	
OID	MIB Group Name	OID	MIB Group Name
1	avCmVersion	25	g3version

1.9	avCmVersionMemConfig	2.7	g3vintageCurMemory
2	**avCmServerAlarm**	**21**	**g3servalm**
3	**avCmConfiguration**	**3**	**g3config**
4 4.1	**avCmServerInfo** avCmServerInfoProductID	**2.17**	g3vintageProductID
5	**avCmAlarms**	**4**	**g3alarms**
6	**avCmErrors**	**5**	**g3errors**
7	**avCmListStation**	**11**	**g3station**
8	**avCmListStationRange**		
9	**avCmUpdate**	**26**	**g3update**
10	**avCmListTrunk**	**32**	**g3trunkgrp**
11	**avCmListTrunkRange**		
12	**avCmCapacityAsai**		
13	**avCmCapacityBcms**		
14	**avCmCapacityCallVec**		
15	**avCmCapacityDialPlan**		
16	**avCmCapacityHuntGrps**		
17	**avCmCapacityAnnounce**		
18	**avCmCapacityTrunks**		
19	**avCmCapacityVoiceTerm**		
20	**avCmCapacityLicGroup**		
21	**avCmCapacityIPUsage**		
21.1	avCmCapacityUsageIPStatUsed	2.35	g3vintageIPStaReg
21.2	avCmCapacityUsageIPStatAvail	2.36	g3vintageIPStaAvail
21.3	avCmCapacityUsageIPStatSysLim	2.37	g3vintageIPStaLimit
21.4	avCmCapacityUsageIPAtConUsed	2.38	g3vintageIPAttdReg
21.5	avCmCapacityUsageIPAtConAvail	2.39	g3vintageIPAttdAvail
21.6	avCmCapacityUsageIPAtConSysLim	2.40	g3vintageIPAttdLimit
21.7	avCmCapacityUsageRemOffStatUsed	2.41	g3vintageRemoteOffReg
21.8	avCmCapacityUsageRemOffStatAvail	**2.42**	g3vintageRemoteOffAvail
21.9	avCmCapacityUsageRemOffStatSysLim	**2.43**	g3vintage RemoteOffLimit
22	**avCmCapacityIPUsage**		
23	**avCmCapacityCurrentUsage**		
24	**avCmListMemTrunk**	**13**	**g3trunkmem**
25	**avCmListMemTrunkRange**		
26	**avCmStatusTrunk**	**14**	**g3trunksta**
27	**avCmStatusTrunkRange**	**154**	**g3trunkstatOptions**
28	**avCmStatusTrunkMem**		
29	**avCmStatusHealth**	**6**	**g3health**
30	**avCmStatusStation**	**12**	**g3statsta**
31	**avCmStatusStationRange**	**155**	**g3statstaOptions**
32	**avCmMeasOccSum**	**50**	**g3occsum**

33	**avCmMeasOccLstHr**		
34	**avCmMeasOccBusIntvl**		
35	**avCmMeasTrkGrpSumLstHr**	**53**	**g3trunksum**
36	**avCmMeasTrkGrpSumToday**		
37	**avCmMeasTrkGrpSumYest**	**54**	**g3pktrunksum**
38	**avCmMeasOutTrkLstHr**	**76**	**g3trunkout**
39	**avCmMeasOutTrkToday**		
40	**avCmMeasOutTrkYest**	**57**	**g3pktrunkout**
41	**avCmMeasLightUseTrkLstHr**	**77**	**g3trunklight**
42	**avCmMeasLightUseTrkToday**		
43	**avCmMeasLightUsedYest**	**58**	**g3ptrunklight**
44	**avCmMeasSecViolationSum**	**73**	**g3secviolate**
45	**avCmListMedGateway**	**104**	**g3gateway**
46	**avCmListConfigMedGateway**	**105**	**g3mediacfg**
47	**avCmStatusMedGateway**	**106**	**g3stamedia**
48	**avCmListIpUnregistered**	**133**	**g3ipunreg**
49	**avCmListSurvProcess**	**153**	**g3lspall**
50	**avCmDispSurvProcess**	**150**	**g3survprocess**
51	**avCmListNodeNamesAll**	**101**	**g3nodename**
52	**avCmListNodeNamesV4**	**151**	**g3nodenameipv4**
53	**avCmListNodeNamesV6**	**152**	**g3nodenameipv6**
54	**avCmListAnn**		
55	**avCmListIpUnregisteredRange**		
56	**avCmDispTime**	**18**	**g3timedate**
57	**avCmMessagingAlarm**	**22**	**g3msgalm**
58	**avCmInitCauses**	**38**	**g3restart**
59	**avCmMeasAnnIntegAllLstHr**	**99**	**g3anninteg**
60	**avCmMeasAnnIntegAllToday**		
61	**avCmMeasAnnIntegAllYest**	**100**	**g3pkanninteg**
62	**avCmStatusMGAnn**	**107**	**g3stamgann**
63	**avCmListIntegAnn**		
64	**avCmMeasIpVoStatHrJitter**	**146**	**g3jitter**
65	**avCmMeasIPVoStatHrPkLos**	**147**	**g3pkloss**
66	**avCmMeasIpVoStatHrRtDel**	**148**	**g3rtdelay**
67	**avCmMeasIpVoStatHrData**		
68	**avCmMeasIpVoStatSumJitterLastHr**		
69	**avCmMeasIpVoStatSumJitterToday**		
70	**avCmMeasIpVoStatSumJitterYest**		
71	**avCmMeasIpVoStatSumPkLosLstHr**		
72	**avCmMeasIpVoStatSumPkLosToday**		
73	**avCmMeasIpVoStatSumPkLosYest**		

74	**avCmMeasIpVoStatSumRtdelLstHr**		
75	**avCmMeasIpVoStatSumRtdelToday**		
76	**avCmMeasIpVoStatSumRtdelYest**		
77	**avCmMeasIpVoStatSumDataLstHr**		
78	**avCmMeasIpVoStatSumDataToday**		
79	**avCmMeasIpVoStatSumDataYest**		
80	**avCmMeasAnnAllLstHr**	**97**	**g3annall**
81	**avCmMeasAnnAllToday**		
82	**avCmMeasAnnAllYest**	**98**	**g3pkannall**
83	**avCmPlatCmds**	**24**	**g3platcmds**
84	**avCmListSigGrp**		
85	**avCmStatusSigGrp**	**37**	**G3siggroup**
86	**avCmListHuntGrp**	**65**	**g3huntlist**
87	**avCmListMeasHuntGrpLstHr**	**63**	**g3hunt**
88	**avCmListMeasHuntGrpToday**		
89	**avCmListMeasHuntGrpYest**	**64**	**g3pkhunt**
90	**avCmListIpInterfaceAll**	**103**	**g3ipinterface**
91	**avCmListInterfaceMedpro**		
92	**avCmStatusMedproBrd**	**132**	**g3statmedpro**
93	**avCmListIpServerIntf**	**108**	**g3ipserver**
94	**avCmStatusLink**	**135**	**g3statlink**
95	**avCmListIPNetRegMonitor**	**92**	**g3ipregion**
96	**avCmStatIpNetRegion**	**130**	**g3statregion**
97	**avCmListMeasToneRecLstHr**	**66**	**g3tonerec**
98	**avCmListMeasToneRecToday**	**new**	
99	**avCmListMeasToneRecYest**	**67**	**g3pktonerec**
100	**avCmListMeasToneRecSumLstHr**	**68**	**g3tonerecsum**
101	**avCmListMeasToneRecSumToday**	**new**	
102	**avCmListMeasToneRecSumYest**	**69**	**g3pktonerecsum**
103 103.5 103.3 103.1	**avCmSysParamCust** avCmSysParamCustPortNetSup avCmSysParamCustLSP avCmSysParamCustPlat	 **2.31** **2.32** **2.33**	 g3vintagePortNetSupport g3vintageLocalSpareProc g3vintagePlatform

Figure 3 – AVAYA-AURA-CM-MIB to G3-MIB OID Conversion Table

The following table outlines the G3-MIB Groups that are not supported by in the AVAYA-AURA-CM-MIB.

G3-MIB **1.3.6.1.4.1.6889.2.8.1**

OID	MIB Group Name
2.2.26	g3vintageACAFlag
7	**g3cabinet**
23	**g3ipevt**
30	**g3ds1cfg**
31	**g3extdev**
34	**g3ds1**
36	**g3pnchealth**
40	**g3routepattern**
41	**g3trunkcfg**
42	**g3stationcfg**
44	**g3aca**
51	**g3attgrp**
52	**g3attpos**
59	**g3pnblock**
60	**g3pkpnblock**
70	**g3rpatcfg**
71	**g3rpat**
72	**g3pkrpat**
74	**g3cbctrunk**
78	**g3loadtotal**
79	**g3loadint**
80	**g3pkloadinc**
81	**g3loadout**
82	**g3loadtan**
83	**g3pkloadtotal**
84	**g3pkloadint**
85	**g3pkloadinc**
86	**g3pkloadout**
87	**g3pkloadtan**
90	**g3ipcodecsum**
91	**g3pkipcodecsum**
93	**g3ipdspsum**
94	**g3pkipdssum**
95	**g3ipsignal**
96	**g3pkipsignal**
110	**g3callratedata**
111	**g3callratevoice**
112	**g3callratsrv**
113	**g3callratemedia**
114	**g3callratetotal**
115	**g3clansocsum**

116	**g3pkclansocsum**
117	**g3clanether**
118	**g3clanppp**
119	**g3mmisum**
120	**g3pkmmisum**
121	**g3esmsum**
122	**g3pkesmsum**
123	**g3voicesum**
124	**g3pkvoicesum**
125	**g3ipintlist**
128	**g3eventhour**
129	**g3eventday**
134	**g3commlink**
136	**g3covpath**
137	**g3pkcovpath**
138	**g3principal**
139	**g3pkprincipal**

Figure 4 – List of AVAYA-AURA-CM-MIB OIDs Not Supported

4.0 Fault Performance Agent.

The FPAgent was replaced in CM Release 6.3.1xx and CM Release 7.0 by the CMFPAgent. The CMFPAgent is used to send notifications messages in the form of trap or inform messages to administered trap destinations. The G3-TRAP-MIB was retired and a new AVAYA-AURA-CMALARM-MIB was developed. The new MIB supports almost 1000 unique trap OIDs based on a combination of MO-Name/Source-Name and Alarm Severity Levels. As with the previous implementation, the CMFPAgent allows users to restrict the type of notifications that can be sent out using the Filters Feature. A new feature, the Alarm Level Adjustment feature, was added in CM Release 6.3.1xx and CM Release 7.0. This feature allows users to change a traps severity level or discard a trap. The below table outlines the CMFPAgents changes:

What has Changed?	OLD FPAgent	NEW CMFPAgent
OIDs	Only a limited number of trap OIDs such as Major, Minor, Warning, Resolved, Cleared Alarm, Test, and Restart were supported in the G3-TRAP-MIB.	Trap OIDs in the AVAYA-AURA-CMALARM-MIB are based on the combination of MO-Name/Source-Name and Severity Level. In addition each Restart level was given a unique trap OID.

OID format/varbinds	Active Communication Manager and Platform alarms use the same format and Resolved Communication Manager Alarms and Platform alarms used the same format	Different trap formats are defined depending on what type of trap is being generated.
Filters Feature	Filters were complex and were difficult to use.	Removed some of the complexity of administering filters by removing the ability to send an individual trap to different IP addresses. This feature was only supported using the SNMP get and SNMP set commands. It was not supported by the SMI Filters Page.
Alarm Level Adjustments Feature		A New feature which allows users to change the severity level of an outgoing trap or discard it.

4.1 Fault Performance Traps

FP notification messages (Traps) were enhanced in CM Release 6.3.1xx and CM Release 7.0 to generate unique OIDs based on the combination of MO-Names/Source-Names and Alarm-Severity. Communication Manager Alarms are generated against Maintenance Objects (MOs) and Platform Alarms are generated against Source Names. CM supports four alarm severities; Major, Minor, Warning and Resolved. Therefore, every MO and Source Name supported by CM has four defined trap types; Major; Minor, Warning, and Resolved. Any Communication Manager alarms that do not map to a defined trap OID will be reported as a generic CommunicaMgr traps - OIDs 4000-4003. Any Platform/Server alarms that do not map to a defined trap OID will be reported as a generic Server traps - OIDs 3000-3003. Messaging alarms were not enhanced and only support generic messaging traps – OIDs 2000-2003. Here is a list of the trap OID ranges supported in the new MIB:

1-1999 - Miscellaneous alarm traps such Cleared Notification and Test Alarm Traps.
1000-1999 – Restart Alarm Notifications.
2000-2999 - Messaging Alarm Notifications.
3000-3999 – Server/Platform Notifications.
4000-XXXX – Communication Manager Notifications.

4.1.1 Communication Manager Alarm Traps

As previously mentioned, Communication Manager Alarm Trap OIDs are based on a combination of MO-Names and Severity Level. Therefore, a Communication Manager TTR-LEV alarm can produce the following four traps; **avCmAlmTtrLevMajor**, **avCmAlmTtrLevMinor, avCmAlmTtrLevWarning**, and **avCmAlmTtrResolved** as defined in the AVAYA-AURA-CMALRM-MIB:

```
-- CommunicaMgr Process TTR-LEV Alarms: 5940-5943

avCmAlmTtrLevMajor    NOTIFICATION-TYPE
   OBJECTS { avCmAlmIPAddress, avCmAlmSystemName, avCmAlmProductID,
             avCmAlmAlarmLoc, avCmAlmMaintName, avCmAlmOnBrd,
             avCmAlmAltName, avCmAlmAlarmSeverity,
             avCmAlmOrigModAlarmSeverity, avCmAlmAlarmedDate,
             avCmAlmAlarmedTime, avCmAlmErrorCodes, avCmAlmNewModFlag }
  STATUS current
  DESCRIPTION " A Major TTR-LEV alarm has been generated by the
               CommunicaMgr process. "
  ::= { avCmAlmNotifications 5940 }

avCmAlmTtrLevMinor    NOTIFICATION-TYPE
   OBJECTS { avCmAlmIPAddress, avCmAlmSystemName, avCmAlmProductID,
             avCmAlmAlarmLoc, avCmAlmMaintName, avCmAlmOnBrd,
             avCmAlmAltName, avCmAlmAlarmSeverity,
             avCmAlmOrigModAlarmSeverity, avCmAlmAlarmedDate,
             avCmAlmAlarmedTime, avCmAlmErrorCodes, avCmAlmNewModFlag }
  STATUS current
  DESCRIPTION " A Minor TTR-LEV alarm has been generated by the
               CommunicaMgr process. "
  ::= { avCmAlmNotifications 5941 }

avCmAlmTtrLevWarning    NOTIFICATION-TYPE
   OBJECTS { avCmAlmIPAddress, avCmAlmSystemName, avCmAlmProductID,
             avCmAlmAlarmLoc, avCmAlmMaintName, avCmAlmOnBrd,
             avCmAlmAltName, avCmAlmAlarmSeverity,
             avCmAlmOrigModAlarmSeverity, avCmAlmAlarmedDate,
             avCmAlmAlarmedTime, avCmAlmErrorCodes, avCmAlmNewModFlag }
  STATUS current
  DESCRIPTION " A Warning TTR-LEV alarm has been generated by the
               CommunicaMgr process. "
  ::= { avCmAlmNotifications 5942 }

avCmAlmTtrLevResolved    NOTIFICATION-TYPE
   OBJECTS { avCmAlmIPAddress, avCmAlmSystemName, avCmAlmProductID,
             avCmAlmAlarmLoc, avCmAlmMaintName, avCmAlmOnBrd,
             avCmAlmAltName, avCmAlmAlarmSeverity,
             avCmAlmOrigModAlarmSeverity, avCmAlmAlarmedDate,
             avCmAlmAlarmedTime, avCmAlmResolvedDate,
             avCmAlmResolvedTime, avCmAlmNewModFlag }
  STATUS current
  DESCRIPTION " A Resolved TTR-LEV alarm has been generated by the
```


CommunicaMgr process. "
::= { avCmAlmNotifications 5943 }

In addition to supporting unique trap OIDs based on the combination of MO Names and Severity Levels, the trap format was modified as well. In the previous SNMP implementation only one active trap format and one resolved trap format was used by both Communication Manager Alarms and Platform Alarms. This caused confusion. Therefore, the new implementation supports different trap formats (varbinds) for Communication Manager Alarms and Platform Alarms.

4.1.1.1 Active Communication Manager Alarm Traps

Active Communication Manager Alarm Traps are generated from alarms with a severity level of Major, Minor, and Warning and support the following varbinds:

avCmAlmIPAddress
avCmAlmSystemName
avCmAlmProductID
avCmAlmAlarmLoc
avCmAlmMaintName
avCmAlmOnBrd
avCmAlmAltName
avCmAlmAlarmSeverity
avCmAlmOrigModAlarmSeverity
avCmAlmAlarmedDate
avCmAlmAlarmedTime
avCmAlmErrorCodes
avCmAlmNewModFlag

4.1.1.1.1 Example of a Major Communication Manager Alarm Trap

Here is an example of an active major Communication Manager SNMP trap:

avCmAlmTtrLevMajor
Message reception date: 5/20/2015
Message reception time: 5:12:54.357 PM
Time stamp: 0 days 00h:56m:57s.89th (341789)
Message type: Notification (Trap)
Protocol version: SNMPv2c
Transport: IP/UDP
Agent
Address: 10.129.178.85
Port: 58925
Manager
Address: 192.168.1.2
Port: 162
Community: public
Bindings (15)
Binding #1: sysUpTimeInstance *** (timeticks) 0 days 00h:56m:57s.89th (341789)

Binding #2: snmpTrapOID.0 *** (OBJECT IDENTIFIER) avCmAlmTtrLevMajor
Binding #3: **avCmAlmIPAddress** *** (SnmpAdminString) 10.129.178.85 [31.30.2E.31.32.39.2E.31.37.38.2E.38.35 (hex)]
Binding #4: **avCmAlmSystemName** *** (SnmpAdminString) dell4snmp-cm [64.65.6C.6C.34.73.6E.6D.70.2D.63.6D (hex)]
Binding #5: **avCmAlmProductID** *** (SnmpAdminString) 1000000000 [31.30.30.30.30.30.30.30.30.30 (hex)]
Binding #6: **avCmAlmAlarmLoc** *** (SnmpAdminString) (zero-length) [(hex)]
Binding #7: **avCmAlmMaintName** *** (SnmpAdminString) TTR_LEV [54.54.52.5F.4C.45.56 (hex)]
Binding #8: **avCmAlmOnBrd** *** (SnmpAdminString) n [6E (hex)]
Binding #9: **avCmAlmAltName** *** (SnmpAdminString) (zero-length) [(hex)]
Binding #10: **avCmAlmAlarmSeverity** *** (SnmpAdminString) MAJ [4D.41.4A (hex)]
Binding #11: **avCmAlmOrigModAlarmSeverity** *** (SnmpAdminString) MAJ [4D.41.4A (hex)]
Binding #12: **avCmAlmAlarmedDate** *** (SnmpAdminString) 00/00 [30.30.2F.30.30 (hex)]
Binding #13: **avCmAlmAlarmedTime** *** (SnmpAdminString) 00:00:00 [30.30.3A.30.30.3A.30.30 (hex)]
Binding #14: **avCmAlmErrorCodes** *** (SnmpAdminString) 18 [31.38 (hex)]
Binding #15: **avCmAlmNewModFlag** *** (SnmpAdminString) N [4E (hex)]

4.1.1.1.2 Example of a Minor Communication Manager Alarm Trap

Here is an example of an active minor Communication Manager SNMP trap:

avCmAlmMedGtwyMinor
Message reception date: 5/20/2015
Message reception time: 5:16:35.787 PM
Time stamp: 0 days 01h:00m:39s.30th (363930)
Message type: Notification (Trap)
Protocol version: SNMPv2c
Transport: IP/UDP
Agent
Address: 10.129.178.85
Port: 58925
Manager
Address: 192.168.1.2
Port: 162
Community: public
Bindings (15)
Binding #1: sysUpTimeInstance *** (timeticks) 0 days 01h:00m:39s.30th (363930)
Binding #2: snmpTrapOID.0 *** (OBJECT IDENTIFIER) avCmAlmMedGtwyMinor
Binding #3: avCmAlmIPAddress *** (SnmpAdminString) 10.129.178.85 [31.30.2E.31.32.39.2E.31.37.38.2E.38.35 (hex)]
Binding #4: avCmAlmSystemName *** (SnmpAdminString) dell4snmp-cm [64.65.6C.6C.34.73.6E.6D.70.2D.63.6D (hex)]
Binding #5: avCmAlmProductID *** (SnmpAdminString) 1000000000 [31.30.30.30.30.30.30.30.30.30 (hex)]
Binding #6: avCmAlmAlarmLoc *** (SnmpAdminString) 074 [30.37.34 (hex)]

Binding #7: avCmAlmMaintName *** (SnmpAdminString) MED-GTWY
[4D.45.44.2D.47.54.57.59 (hex)]
Binding #8: avCmAlmOnBrd *** (SnmpAdminString) n [6E (hex)]
Binding #9: avCmAlmAltName *** (SnmpAdminString) (zero-length) [(hex)]
Binding #10: avCmAlmAlarmSeverity *** (SnmpAdminString) MIN [4D.49.4E (hex)]
Binding #11: avCmAlmOrigModAlarmSeverity *** (SnmpAdminString) MAJ-MIN
[4D.41.4A.2D.4D.49.4E (hex)]
Binding #12: avCmAlmAlarmedDate *** (SnmpAdminString) 09/03 [30.39.2F.30.33 (hex)]
Binding #13: avCmAlmAlarmedTime *** (SnmpAdminString) 17:56:57
[31.37.3A.35.36.3A.35.37 (hex)]
Binding #14: avCmAlmErrorCodes *** (SnmpAdminString) none [6E.6F.6E.65 (hex)]
Binding #15: avCmAlmNewModFlag *** (SnmpAdminString) M [4D (hex)]

4.1.1.1.3 Example of a Communication Manager Warning Alarm Trap

Here is an example of an active warning Communication Manager SNMP trap:

avCmAlmMedGtwyWarning
Message reception date: 5/22/2015
Message reception time: 10:30:15.469 AM
Time stamp: 1 days 00h:23m:17s.44th (8779744)
Message type: Notification (Trap)
Protocol version: SNMPv2c
Transport: IP/UDP
Agent
Address: 10.129.178.85
Port: 42255
Manager
Address: 192.168.1.2
Port: 162
Community: public
Bindings (15)
Binding #1: sysUpTimeInstance *** (timeticks) 1 days 00h:23m:17s.44th (8779744)
Binding #2: snmpTrapOID.0 *** (OBJECT IDENTIFIER) avCmAlmMedGtwyWarning
Binding #3: avCmAlmIPAddress *** (SnmpAdminString) 10.129.178.85
[31.30.2E.31.32.39.2E.31.37.38.2E.38.35 (hex)]
Binding #4: avCmAlmSystemName *** (SnmpAdminString) snmp-mehm
[64.65.6C.6C.34.73.6E.6D.70.2D.6D.65.68.6D (hex)]
Binding #5: avCmAlmProductID *** (SnmpAdminString) 1000000000
[31.30.30.30.30.30.30.30.30.30 (hex)]
Binding #6: avCmAlmAlarmLoc *** (SnmpAdminString) 074 [30.37.34 (hex)]
Binding #7: avCmAlmMaintName *** (SnmpAdminString) MED-GTWY
[4D.45.44.2D.47.54.57.59 (hex)]
Binding #8: avCmAlmOnBrd *** (SnmpAdminString) n [6E (hex)]
Binding #9: avCmAlmAltName *** (SnmpAdminString) (zero-length) [(hex)]
Binding #10: avCmAlmAlarmSeverity *** (SnmpAdminString) WRN [57.52.4E (hex)]
Binding #11: avCmAlmOrigModAlarmSeverity *** (SnmpAdminString) MIN-WRN
[4D.49.4E.2D.57.52.4E (hex)]
Binding #12: avCmAlmAlarmedDate *** (SnmpAdminString) 09/03 [30.39.2F.30.33 (hex)]
Binding #13: avCmAlmAlarmedTime *** (SnmpAdminString) 17:56:57
[31.37.3A.35.36.3A.35.37 (hex)]

Binding #14: avCmAlmErrorCodes *** (SnmpAdminString) 769 [37.36.39 (hex)]
Binding #15: avCmAlmNewModFlag *** (SnmpAdminString) N [4E (hex)]

4.1.1.2 **Resolved Communication Manager Alarm Traps**

Resolved Communication Manager Alarm Traps are generated from resolved alarms and have a different format than active traps as follows:

avCmAlmIPAddress
avCmAlmSystemName
avCmAlmProductID
avCmAlmAlarmLoc
avCmAlmMaintName
avCmAlmOnBrd
avCmAlmAltName
avCmAlmAlarmSeverity
avCmAlmOrigModAlarmSeverity
avCmAlmAlarmedDate
avCmAlmAlarmedTime
avCmAlmResolvedDate
avCmAlmResolvedTime
avCmAlmNewModFlag

Here is an example of a resolved Communication Manager Alarm Trap:

avCmAlmMedGtwyResolved
Message reception date: 5/20/2015
Message reception time: 6:04:45.618 PM
Time stamp: 0 days 00h:03m:52s.96th (23296)
Message type: Notification (Trap)
Protocol version: SNMPv2c
Transport: IP/UDP
Agent
Address: 10.129.178.85
Port: 57561
Manager
Address: 192.168.1.2
Port: 162
Community: public
Bindings (16)
Binding #1: sysUpTimeInstance *** (timeticks) 0 days 00h:03m:52s.96th (23296)
Binding #2: snmpTrapOID.0 *** (OBJECT IDENTIFIER) avCmAlmMedGtwyResolved
Binding #3: **avCmAlmIPAddress** *** (SnmpAdminString) 10.129.178.85
[31.30.2E.31.32.39.2E.31.37.38.2E.38.35 (hex)]
Binding #4: **avCmAlmSystemName** *** (SnmpAdminString) snmp-mehm
[64.65.6C.6C.34.73.6E.6D.70.2D.6D.65.68.6D (hex)]
Binding #5: **avCmAlmProductID** *** (SnmpAdminString) 1000000000
[31.30.30.30.30.30.30.30.30.30 (hex)]
Binding #6: **avCmAlmAlarmLoc** *** (SnmpAdminString) 074 [30.37.34 (hex)]

Binding #7: a**vCmAlmMaintName** *** (SnmpAdminString) MED-GTWY [4D.45.44.2D.47.54.57.59 (hex)]
Binding #8: a**vCmAlmOnBrd** *** (SnmpAdminString) n [6E (hex)]
Binding #9: **avCmAlmAltName** *** (SnmpAdminString) (zero-length) [(hex)]
Binding #10: **avCmAlmAlarmSeverity** *** (SnmpAdminString) MIN [4D.49.4E (hex)]
Binding #11: **avCmAlmOrigModAlarmSeverity** *** (SnmpAdminString) MAJ-MIN [4D.41.4A.2D.4D.49.4E (hex)]
Binding #12: avCmAlmAlarmedDate *** (SnmpAdminString) 09/03 [30.39.2F.30.33 (hex)]
Binding #13: **avCmAlmAlarmedTime** *** (SnmpAdminString) 17:57:57 [31.37.3A.35.37.3A.35.37 (hex)]
Binding #14: a**vCmAlmResolvedDate** *** (SnmpAdminString) 09/03 [30.39.2F.30.33 (hex)]
Binding #15: **avCmAlmResolvedTime** *** (SnmpAdminString) 18:02:14 [31.38.3A.30.32.3A.31.34 (hex)]
Binding #16: **avCmAlmNewModFlag** *** (SnmpAdminString) N [4E (hex)]

4.1.2 Platform/Server Alarm Traps

As previously mentioned, Platform/Server Alarm Traps have unique trap OIDs based on the combination of Source Name and Severity Level. In addition, the Platform/Server trap format was modified to include only those varbinds used by Platform/Server traps. Thus, a SVC_MON Platform alarm can generate the following four traps; **avCmAlmServSvcMonMajor**, **avCmAlmServSvcMonMinor,** **avCmAlmServSvcMonWarning**, and **avCmAlmServSvcMonResolved** as defined in the AVAYA-AURA-CMALARM-MIB:

```
-- SVC_MON Server Alarms: 3230-3233

avCmAlmServSvcMonMajor    NOTIFICATION-TYPE
   OBJECTS { avCmAlmIPAddress, avCmAlmSystemName, avCmAlmProductID,
             avCmAlmServSourceName, avCmAlmServEvtID,
             avCmAlmAlarmSeverity, avCmAlmOrigModAlarmSeverity,
             avCmAlmAlarmedDate, avCmAlmAlarmedTime,
             avCmAlmServLogID, avCmAlmServAlarmDescription }
   STATUS current
   DESCRIPTION " A Major SVC_MON Server alarm has been generated by the
                 system. "
   ::= { avCmAlmNotifications 3230 }

avCmAlmServSvcMonMinor    NOTIFICATION-TYPE
   OBJECTS { avCmAlmIPAddress, avCmAlmSystemName, avCmAlmProductID,
             avCmAlmServSourceName, avCmAlmServEvtID,
             avCmAlmAlarmSeverity, avCmAlmOrigModAlarmSeverity,
             avCmAlmAlarmedDate, avCmAlmAlarmedTime,
             avCmAlmServLogID, avCmAlmServAlarmDescription }
   STATUS current
   DESCRIPTION " A Minor SVC_MON Server alarm has been generated by the
                 system. "
   ::= { avCmAlmNotifications 3231 }

avCmAlmServSvcMonWarning    NOTIFICATION-TYPE
```



```
   OBJECTS { avCmAlmIPAddress, avCmAlmSystemName, avCmAlmProductID,
          avCmAlmServSourceName, avCmAlmServEvtID,
          avCmAlmAlarmSeverity, avCmAlmOrigModAlarmSeverity,
          avCmAlmAlarmedDate, avCmAlmAlarmedTime,
          avCmAlmServLogID, avCmAlmServAlarmDescription }
  STATUS current
  DESCRIPTION " A Warning SVC_MON Server alarm has been generated by the
           system. "
  ::= { avCmAlmNotifications 3232 }

avCmAlmServSvcMonResolved    NOTIFICATION-TYPE
   OBJECTS { avCmAlmIPAddress, avCmAlmSystemName, avCmAlmProductID,
          avCmAlmServSourceName, avCmAlmServEvtID,
          avCmAlmAlarmSeverity, avCmAlmOrigModAlarmSeverity,
          avCmAlmResolvedDate, avCmAlmResolvedTime,
          avCmAlmServLogID, avCmAlmServAlarmDescription }
  STATUS current
  DESCRIPTION " A SVC_MON Server alarm has been resolved by the system. "
  ::= { avCmAlmNotifications 3233 }
```

4.1.2.1 Active Platform/Server Alarm Traps

Active Platform/Server Alarm Traps are generated from alarms with a severity level of Major, Minor, and Warning and support the following varbinds:

avCmAlmIPAddress
avCmAlmSystemName
avCmAlmProductID
avCmAlmServSourceName
avCmAlmServEvtID
avCmAlmAlarmSeverity
avCmAlmOrigModAlarmSeverity
avCmAlmAlarmedDate
avCmAlmAlarmedTime
avCmAlmServLogID
avCmAlmServAlarmDescription

4.1.2.1.1 Example of a Major Platform/Server Alarm Trap

Here is an example of a major Platform/Server Alarm Trap

```
avCmAlmServSvcMonMajor
  Message reception date: 5/22/2015
  Message reception time: 11:41:30.756 AM
  Time stamp: 1 days 01h:34m:33s.95th (9207395)
  Message type: Notification (Trap)
  Protocol version: SNMPv2c
  Transport: IP/UDP
  Agent
    Address: 10.129.178.85
```


Port: 42255
Manager
Address: 192.168.1.2
Port: 162
Community: public
Bindings (13)
Binding #1: sysUpTimeInstance *** (timeticks) 1 days 01h:34m:33s.95th (9207395)
Binding #2: snmpTrapOID.0 *** (OBJECT IDENTIFIER) avCmAlmServSvcMonMajor
Binding #3: avCmAlmIPAddress *** (SnmpAdminString) 10.129.178.85
[31.30.2E.31.32.39.2E.31.37.38.2E.38.35 (hex)]
Binding #4: avCmAlmSystemName *** (SnmpAdminString) snmp-mehm
[64.65.6C.6C.34.73.6E.6D.70.2D.6D.65.68.6D (hex)]
Binding #5: avCmAlmProductID *** (SnmpAdminString) 1000000000
[31.30.30.30.30.30.30.30.30.30 (hex)]
Binding #6: avCmAlmServSourceName *** (SnmpAdminString) SVC_MON
[53.56.43.5F.4D.4F.4E (hex)]
Binding #7: avCmAlmServEvtID *** (SnmpAdminString) 5 [35 (hex)]
Binding #8: avCmAlmAlarmSeverity *** (SnmpAdminString) MAJ [4D.41.4A (hex)]
Binding #9: avCmAlmOrigModAlarmSeverity *** (SnmpAdminString) MAJ [4D.41.4A
(hex)]
Binding #10: avCmAlmAlarmedDate *** (SnmpAdminString) 05/22 [30.35.2F.32.32 (hex)]
Binding #11: avCmAlmAlarmedTime *** (SnmpAdminString) 09:41:20
[30.39.3A.34.31.3A.32.30 (hex)]
Binding #12: avCmAlmServLogID *** (SnmpAdminString) A [41 (hex)]
Binding #13: avCmAlmServAlarmDescription *** (SnmpAdminString) service xinetd does
not exist
[73.65.72.76.69.63.65.20.78.69.6E.65.74.64.20.64.6F.65.73.20.6E.6F.74.20.65.78.69.73.74
(hex)]

4.1.2.1.2 Example of a Minor Platform/Server Alarm Trap

Here is an example of a minor Platform/Server Alarm Trap

avCmAlmServSvcMonMinor
Message reception date: 5/20/2015
Message reception time: 5:03:30.153 PM
Time stamp: 0 days 00h:47m:33s.71th (285371)
Message type: Notification (Trap)
Protocol version: SNMPv2c
Transport: IP/UDP
Agent
Address: 10.129.178.85
Port: 58925
Manager
Address: 192.168.1.2
Port: 162
Community: public
Bindings (13)
Binding #1: sysUpTimeInstance *** (timeticks) 0 days 00h:47m:33s.71th (285371)
Binding #2: snmpTrapOID.0 *** (OBJECT IDENTIFIER) avCmAlmServSvcMonMinor

Binding #3: avCmAlmIPAddress *** (SnmpAdminString) 10.129.178.85 [31.30.2E.31.32.39.2E.31.37.38.2E.38.35 (hex)]
Binding #4: avCmAlmSystemName *** (SnmpAdminString) dell4snmp-cm [64.65.6C.6C.34.73.6E.6D.70.2D.63.6D (hex)]
Binding #5: avCmAlmProductID *** (SnmpAdminString) 1000000000 [31.30.30.30.30.30.30.30.30.30 (hex)]
Binding #6: avCmAlmServSourceName *** (SnmpAdminString) SVC_MON [53.56.43.5F.4D.4F.4E (hex)]
Binding #7: avCmAlmServEvtID *** (SnmpAdminString) 2 [32 (hex)]
Binding #8: avCmAlmAlarmSeverity *** (SnmpAdminString) MIN [4D.49.4E (hex)]
Binding #9: avCmAlmOrigModAlarmSeverity *** (SnmpAdminString) MIN [4D.49.4E (hex)]
Binding #10: avCmAlmAlarmedDate *** (SnmpAdminString) 05/20 [30.35.2F.32.30 (hex)]
Binding #11: avCmAlmAlarmedTime *** (SnmpAdminString) 15:03:21 [31.35.3A.30.33.3A.32.31 (hex)]
Binding #12: avCmAlmServLogID *** (SnmpAdminString) A [41 (hex)]
Binding #13: avCmAlmServAlarmDescription *** (SnmpAdminString) service atd could not be restarted [73.65.72.76.69.63.65.20.61.74.64.20.63.6F.75.6C.64.20.6E.6F.74.20.62.65.20.72.65.73.74.61.72.74.65.64 (hex)]

4.1.2.1.2 Example of a Warning Platform/Server Alarm Trap

Here is an example of a warning Platform/Server Alarm Trap:

avCmAlmServPruneWarning
Message reception date: 5/22/2015
Message reception time: 11:57:11.279 AM
Time stamp: 1 days 01h:50m:14s.41th (9301441)
Message type: Notification (Trap)
Protocol version: SNMPv2c
Transport: IP/UDP
Agent
Address: 10.129.178.85
Port: 42255
Manager
Address: 192.168.1.2
Port: 162
Community: public
Bindings (13)
Binding #1: sysUpTimeInstance *** (timeticks) 1 days 01h:50m:14s.41th (9301441)
Binding #2: snmpTrapOID.0 *** (OBJECT IDENTIFIER) avCmAlmServPruneWarning
Binding #3: avCmAlmIPAddress *** (SnmpAdminString) 10.129.178.85 [31.30.2E.31.32.39.2E.31.37.38.2E.38.35 (hex)]
Binding #4: avCmAlmSystemName *** (SnmpAdminString) snmp-mehm [64.65.6C.6C.34.73.6E.6D.70.2D.6D.65.68.6D (hex)]
Binding #5: avCmAlmProductID *** (SnmpAdminString) 1000000000 [31.30.30.30.30.30.30.30.30.30 (hex)]
Binding #6: avCmAlmServSourceName *** (SnmpAdminString) PRUNE [50.52.55.4E.45 (hex)]

Binding #7: avCmAlmServEvtID *** (SnmpAdminString) 11 [31.31 (hex)]
Binding #8: avCmAlmAlarmSeverity *** (SnmpAdminString) WRN [57.52.4E (hex)]
Binding #9: avCmAlmOrigModAlarmSeverity *** (SnmpAdminString) WRN [57.52.4E (hex)]
Binding #10: avCmAlmAlarmedDate *** (SnmpAdminString) 05/22 [30.35.2F.32.32 (hex)]
Binding #11: avCmAlmAlarmedTime *** (SnmpAdminString) 09:57:00 [30.39.3A.35.37.3A.30.30 (hex)]
Binding #12: avCmAlmServLogID *** (SnmpAdminString) A [41 (hex)]
Binding #13: avCmAlmServAlarmDescription *** (SnmpAdminString) Removed files under /tmp [52.65.6D.6F.76.65.64.20.66.69.6C.65.73.20.75.6E.64.65.72.20.2F.74.6D.70 (hex)]

4.1.2.1 Resolved Platform/Server Alarm Traps

Resolved Platform Alarm Traps support the following varbinds:

avCmAlmIPAddress
avCmAlmSystemName
avCmAlmProductID
avCmAlmServSourceName
avCmAlmServEvtID
avCmAlmAlarmSeverity,
avCmAlmOrigModAlarmSeverity
avCmAlmResolvedDate
avCmAlmResolvedTime
avCmAlmServLogID,
avCmAlmServAlarmDescription

4.1.2.1.1 Example of a Resolved Platform/Server Alarm Trap

Here is an example of a resolved Platform/Server Trap:

avCmAlmServLxResolved
Message reception date: 5/20/2015
Message reception time: 4:23:50.637 PM
Time stamp: 0 days 00h:07m:54s.33th (47433)
Message type: Notification (Trap)
Protocol version: SNMPv2c
Transport: IP/UDP
Agent
Address: 10.129.178.85
Port: 58925
Manager
Address: 192.168.1.2
Port: 162
Community: public
Bindings (13)
Binding #1: sysUpTimeInstance *** (timeticks) 0 days 00h:07m:54s.33th (47433)
Binding #2: snmpTrapOID.0 *** (OBJECT IDENTIFIER) avCmAlmServLxResolved
Binding #3: **avCmAlmIPAddress** *** (SnmpAdminString) 10.129.178.85 [31.30.2E.31.32.39.2E.31.37.38.2E.38.35 (hex)]

Binding #4: **avCmAlmSystemName** *** (SnmpAdminString) dell4snmp-cm [64.65.6C.6C.34.73.6E.6D.70.2D.63.6D (hex)]
Binding #5: **avCmAlmProductID** *** (SnmpAdminString) 1000000000 [31.30.30.30.30.30.30.30.30.30 (hex)]
Binding #6: **avCmAlmServSourceName** *** (SnmpAdminString) _LX [5F.4C.58 (hex)]
Binding #7: **avCmAlmServEvtID** *** (SnmpAdminString) 5 [35 (hex)]
Binding #8: **avCmAlmAlarmSeverity** *** (SnmpAdminString) WRN [57.52.4E (hex)]
Binding #9: **avCmAlmOrigModAlarmSeverity** *** (SnmpAdminString) WRN [57.52.4E (hex)]
Binding #10: **avCmAlmResolvedDate** *** (SnmpAdminString) 05/20 [30.35.2F.32.30 (hex)]
Binding #11: **avCmAlmResolvedTime** *** (SnmpAdminString) 14:23:40 [31.34.3A.32.33.3A.34.30 (hex)]
Binding #12: **avCmAlmServLogID** *** (SnmpAdminString) A [41 (hex)]
Binding #13: **avCmAlmServAlarmDescription** *** (SnmpAdminString) corevector clear [63.6F.72.65.76.65.63.74.6F.72.20.63.6C.65.61.72 (hex)]

4.1.3 Restart Notification Traps

Restart Notification Alarm Traps were also enhanced to support a unique OID for each restart type reported by CM. The AVAYA-AURA-CMALARM-MIB defines five restart traps:

avCmAlmWarmRestart,
avCmAlmCold2Restart
avCmAlmRebootRestart
avCmAlmCoolRestart,
avCmAlmUnknownRestart

Any Restart Notification Alarm that is not defined in the AVAYA-AURA-CMALARM-MIB will be processed in the CMFPAgent as an avCmAlmUnknownRestart trap. Restart Alarm Traps support the following varbinds:

avCmAlmIPAddress,
avCmAlmSystemName
avCmAlmProductID
avCmAlmRestartDateTime
avCmAlmRestartLevel
avCmAlmRestartServer,
avCmAlmRestartCraftDemand
avCmAlmRestartEscalated,
avCmAlmRestartInterchange
avCmAlmRestartCause
avCmAlmRestartRelease
avCmAlmRestartUpdateID

4.1.3.1 Example of a WarmRestart Trap

Here is an example of an avCmAlmWarmRestart:

avCmAlmWarmRestart
Message reception date: 5/20/2015
Message reception time: 4:42:00.065 PM
Time stamp: 0 days 00h:26m:03s.69th (156369)
Message type: Notification (Trap)
Protocol version: SNMPv2c
Transport: IP/UDP
Agent
Address: 10.129.178.85
Port: 58925
Manager
Address: 192.168.1.2
Port: 162
Community: public
Bindings (14)
Binding #1: sysUpTimeInstance *** (timeticks) 0 days 00h:26m:03s.69th (156369)
Binding #2: snmpTrapOID.0 *** (OBJECT IDENTIFIER) avCmAlmWarmRestart
Binding #3: avCmAlmIPAddress *** (SnmpAdminString) 10.129.178.85
[31.30.2E.31.32.39.2E.31.37.38.2E.38.35 (hex)]
Binding #4: avCmAlmSystemName *** (SnmpAdminString) dell4snmp-cm
[64.65.6C.6C.34.73.6E.6D.70.2D.63.6D (hex)]
Binding #5: avCmAlmProductID *** (SnmpAdminString) 1000000000
[31.30.30.30.30.30.30.30.30.30 (hex)]
Binding #6: avCmAlmRestartDateTime *** (SnmpAdminString) 05/20 14:41
[30.35.2F.32.30.20.31.34.3A.34.31 (hex)]
Binding #7: avCmAlmRestartLevel *** (SnmpAdminString) WARM [57.41.52.4D (hex)]
Binding #8: avCmAlmRestartServer *** (SnmpAdminString) A [41 (hex)]
Binding #9: avCmAlmRestartCraftDemand *** (SnmpAdminString) Y [59 (hex)]
Binding #10: avCmAlmRestartEscalated *** (SnmpAdminString) N [4E (hex)]
Binding #11: avCmAlmRestartInterchange *** (SnmpAdminString) N [4E (hex)]
Binding #12: avCmAlmRestartCause *** (SnmpAdminString) Craft Request
[43.72.61.66.74.20.52.65.71.75.65.73.74 (hex)]
Binding #13: avCmAlmRestartRelease *** (SnmpAdminString) R017x.00.0.438.0
[52.30.31.37.78.2E.30.30.2E.30.2E.34.33.38.2E.30 (hex)]
Binding #14: avCmAlmRestartUpdateID *** (SnmpAdminString) 00.0.438.0-777777
[30.30.2E.30.2E.34.33.38.2E.30.2D.37.37.37.37.37.37 (hex)]

4.1.3.2 Example of a Cold2Restart Trap

Here is an example of an avCmCold2Restart Trap:

avCmAlmCold2Restart
Message reception date: 5/20/2015
Message reception time: 4:31:12.428 PM
Time stamp: 0 days 00h:15m:16s.11th (91611)
Message type: Notification (Trap)
Protocol version: SNMPv2c
Transport: IP/UDP
Agent

Address: 10.129.178.85
Port: 58925
Manager
Address: 192.168.1.2
Port: 162
Community: public
Bindings (14)
Binding #1: sysUpTimeInstance *** (timeticks) 0 days 00h:15m:16s.11th (91611)
Binding #2: snmpTrapOID.0 *** (OBJECT IDENTIFIER) avCmAlmCold2Restart
Binding #3: avCmAlmIPAddress *** (SnmpAdminString) 10.129.178.85
[31.30.2E.31.32.39.2E.31.37.38.2E.38.35 (hex)]
Binding #4: avCmAlmSystemName *** (SnmpAdminString) dell4snmp-cm
[64.65.6C.6C.34.73.6E.6D.70.2D.63.6D (hex)]
Binding #5: avCmAlmProductID *** (SnmpAdminString) 1000000000
[31.30.30.30.30.30.30.30.30.30 (hex)]
Binding #6: avCmAlmRestartDateTime *** (SnmpAdminString) 05/20 14:31
[30.35.2F.32.30.20.31.34.3A.33.31 (hex)]
Binding #7: avCmAlmRestartLevel *** (SnmpAdminString) COLD2 [43.4F.4C.44.32 (hex)]
Binding #8: avCmAlmRestartServer *** (SnmpAdminString) A [41 (hex)]
Binding #9: avCmAlmRestartCraftDemand *** (SnmpAdminString) Y [59 (hex)]
Binding #10: avCmAlmRestartEscalated *** (SnmpAdminString) N [4E (hex)]
Binding #11: avCmAlmRestartInterchange *** (SnmpAdminString) N [4E (hex)]
Binding #12: avCmAlmRestartCause *** (SnmpAdminString) Craft Request
[43.72.61.66.74.20.52.65.71.75.65.73.74 (hex)]
Binding #13: avCmAlmRestartRelease *** (SnmpAdminString) R017x.00.0.438.0
[52.30.31.37.78.2E.30.30.2E.30.2E.34.33.38.2E.30 (hex)]
Binding #14: avCmAlmRestartUpdateID *** (SnmpAdminString) 00.0.438.0-777777
[30.30.2E.30.2E.34.33.38.2E.30.2D.37.37.37.37.37.37 (hex)]

4.1.3.3 Example of a RebootRestart Trap

Here is an example of an avCmAlmRebootRestart Trap:

avCmAlmRebootRestart
Message reception date: 5/20/2015
Message reception time: 4:23:50.700 PM
Time stamp: 0 days 00h:07m:54s.40th (47440)
Message type: Notification (Trap)
Protocol version: SNMPv2c
Transport: IP/UDP
Agent
Address: 10.129.178.85
Port: 58925
Manager
Address: 192.168.1.2
Port: 162
Community: public
Bindings (14)
Binding #1: sysUpTimeInstance *** (timeticks) 0 days 00h:07m:54s.40th (47440)
Binding #2: snmpTrapOID.0 *** (OBJECT IDENTIFIER) avCmAlmRebootRestart

Binding #3: avCmAlmIPAddress *** (SnmpAdminString) 10.129.178.85
[31.30.2E.31.32.39.2E.31.37.38.2E.38.35 (hex)]
Binding #4: avCmAlmSystemName *** (SnmpAdminString) dell4snmp-cm
[64.65.6C.6C.34.73.6E.6D.70.2D.63.6D (hex)]
Binding #5: avCmAlmProductID *** (SnmpAdminString) 1000000000
[31.30.30.30.30.30.30.30.30.30 (hex)]
Binding #6: avCmAlmRestartDateTime *** (SnmpAdminString) 05/20 14:23
[30.35.2F.32.30.20.31.34.3A.32.33 (hex)]
Binding #7: avCmAlmRestartLevel *** (SnmpAdminString) REBOOT [52.45.42.4F.4F.54
(hex)]
Binding #8: avCmAlmRestartServer *** (SnmpAdminString) A [41 (hex)]
Binding #9: avCmAlmRestartCraftDemand *** (SnmpAdminString) Y [59 (hex)]
Binding #10: avCmAlmRestartEscalated *** (SnmpAdminString) N [4E (hex)]
Binding #11: avCmAlmRestartInterchange *** (SnmpAdminString) N [4E (hex)]
Binding #12: avCmAlmRestartCause *** (SnmpAdminString) Craft Request
[43.72.61.66.74.20.52.65.71.75.65.73.74 (hex)]
Binding #13: avCmAlmRestartRelease *** (SnmpAdminString) R017x.00.0.438.0
[52.30.31.37.78.2E.30.30.2E.30.2E.34.33.38.2E.30 (hex)]
Binding #14: avCmAlmRestartUpdateID *** (SnmpAdminString) 00.0.438.0-777777
[30.30.2E.30.2E.34.33.38.2E.30.2D.37.37.37.37.37.37 (hex)]

4.1.4 FP Test Alarm Trap

The Test Alarm Trap format was modified to support the new AVAYA-AURA-ALARM-MIB as follows:

```
avCmAlmAlarmTest    NOTIFICATION-TYPE
   OBJECTS { avCmAlmIPAddress, avCmAlmSystemName, avCmAlmProductID,
          avCmAlmAlarmedDate, avCmAlmAlarmedTime, avCmAlmClearDescription }
   STATUS current
   DESCRIPTION " A Test Alarm has been issued by the
             CommunicaMgr process to generate a trap for testing. "

    ::= { avCmAlmNotifications 9 }
```

4.1.4.1. Example of a FP Test Alarm Trap

Here is an example of a SNMPv2c FP Test Trap:

```
avCmAlmAlarmTest
  Message reception date: 5/20/2015
  Message reception time: 4:16:11.596 PM
  Time stamp: 0 days 00h:00m:15s.32th (1532)
  Message type: Notification (Trap)
  Protocol version: SNMPv2c
  Transport: IP/UDP
  Agent
    Address: 10.129.178.85
    Port: 58925
  Manager
```



```
    Address: 192.168.1.2
    Port: 162
  Community: public
  Bindings (8)
    Binding #1: sysUpTimeInstance *** (timeticks) 0 days 00h:00m:15s.32th (1532)
    Binding #2: snmpTrapOID.0 *** (OBJECT IDENTIFIER) avCmAlmAlarmTest
    Binding #3: avCmAlmIPAddress *** (SnmpAdminString) 10.129.178.85
[31.30.2E.31.32.39.2E.31.37.38.2E.38.35 (hex)]
    Binding #4: avCmAlmSystemName *** (SnmpAdminString) dell4snmp-cm
[64.65.6C.6C.34.73.6E.6D.70.2D.63.6D (hex)]
    Binding #5: avCmAlmProductID *** (SnmpAdminString) 1000000000
[31.30.30.30.30.30.30.30.30.30 (hex)]
    Binding #6: avCmAlmAlarmedDate *** (SnmpAdminString) 05/20 [30.35.2F.32.30 (hex)]
    Binding #7: avCmAlmAlarmedTime *** (SnmpAdminString) 14:16:03
[31.34.3A.31.36.3A.30.33 (hex)]
    Binding #8: avCmAlmTestDescription *** (SnmpAdminString) CUSTOMER ALARM TEST
[43.55.53.54.4F.4D.45.52.20.41.4C.41.52.4D.20.54.45.53.54 (hex)]
```

4.1.5 FP Cleared Alarm Notification Trap

The Cleared Alarm Notification Trap format was modified to support the new AVAYA-AURA-ALARM-MIB as follows:

```
avCmAlmAlarmClear    NOTIFICATION-TYPE
   OBJECTS { avCmAlmIPAddress, avCmAlmSystemName, avCmAlmProductID,
          avCmAlmAlarmedDate, avCmAlmAlarmedTime, avCmAlmClearDescription }
   STATUS current
   DESCRIPTION " A Clear Alarm Notification Alarm has been issued by the
              CommunicaMgr process indicating that all previously
              acknowledged alarms have been cleared. "
    ::= { avCmAlmNotifications 8 }
```

4.1.5.1 Example of a Cleared Alarm Notification Trap

Here is an example of a SNMPv2c Cleared Alarm Notification Trap:

```
avCmAlmAlarmClear
  Message reception date: 5/29/2015
  Message reception time: 2:49:42.559 PM
  Time stamp: 0 days 00h:00m:51s.60th (5160)
  Message type: Notification (Trap)
  Protocol version: SNMPv2c
  Transport: IP/UDP
  Agent
    Address: 10.129.178.85
    Port: 44161
  Manager
    Address: 192.168.1.2
    Port: 162
  Community: public
```



```
  Bindings (8)
    Binding #1: sysUpTimeInstance *** (timeticks) 0 days 00h:00m:51s.60th (5160)
    Binding #2: snmpTrapOID.0 *** (OBJECT IDENTIFIER) avCmAlmAlarmClear
    Binding #3: avCmAlmIPAddress *** (SnmpAdminString) 10.129.178.85
[31.30.2E.31.32.39.2E.31.37.38.2E.38.35 (hex)]
    Binding #4: avCmAlmSystemName *** (SnmpAdminString) snmp-mehm
[64.65.6C.6C.34.73.6E.6D.70.2D.6D.65.68.6D (hex)]
    Binding #5: avCmAlmProductID *** (SnmpAdminString) 1000000000
[31.30.30.30.30.30.30.30.30.30 (hex)]
    Binding #6: avCmAlmAlarmedDate *** (SnmpAdminString) 06/28 [30.36.2F.32.38 (hex)]
    Binding #7: avCmAlmAlarmedTime *** (SnmpAdminString) 09:31:54
[30.39.3A.33.31.3A.35.34 (hex)]
    Binding #8: avCmAlmClearDescription *** (SnmpAdminString) ALL ALARMS
RESOLVED  [41.4C.4C.20.41.4C.41.52.4D.53.20.52.45.53.4F.4C.56.45.44.20 (hex)]
```

4.2 Filters Features

SNMP filters play a very important role in CM Alarming and CM SNMP Traps. Filters in CM are positive, meaning you must have at least one filter if you want to send a trap out. Fewer filters are needed when filters are configured to be less restrictive. Accordingly, more filters are needed when filters are configured to be more restrictive. When overlapping filters are configured the least restrictive filter is applied. CM supports three different types of alarm filters: Communication Manager Alarm Filters, Platform/Server Alarm Filters, and Restart Notification Alarm Filters. Communication Manager Alarm Filters can be added, displayed, changed, and deleted via the CM SNMP *FP Filters* SMI Page or using SNMP get and set commands. Restart Alarm Filters and Platform/Server Alarm Filters can only be added, displayed, changed, and deleted using SNMP get and set commands. The previous implementation allowed users to configure individual filters to separate IP addresses using SNMP get and SNMP set commands. This feature is not supported in the new CMFPAgent. Administered filters apply to all configured IP trap destinations.

4.2.1 Default Filters

Default Communication Manager, Platform/Server, and Restart Alarm Filters are added when the system is installed. A snmpwalk of the avCmAlmCmFiltTable , the avCmAlmPlatFiltTable, and, the avCmAlmRstFiltTable can be used to retrieve a newly installed Server's default filters as follows:

```
Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85 avCmAlmCmFiltTable
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltActive.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltResolved.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMajor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMinor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltWarning.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMediaGateway.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltCabinet.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltBoardNumber.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltPort.0 = STRING: -
```


AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltCategory.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMaintenanceObject.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltExtension.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltTrunkGroup.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltTrunkMember.0 = INTEGER: 0

The above default Communication Manager Alarm Filter allows all Active Major and Minor Communication Manager traps to be sent out.

Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c private 10.129.178.85 avCmAlmPlatFiltTable
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltActive.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltResolved.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltMajor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltMinor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltWarning.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltAlarmType.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltAlarmSource.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltEventID.0 = STRING: -

The above default Platform/Server Alarm Filter allows all Active and Resolved Major, Minor, and warning Platform/Server traps to be sent out.

Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c private 10.129.178.85 avCmAlmRstFiltTable
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltWarmRestart.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltCold2Restart.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltReboot.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltCoolRestart.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltInterchange.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltCraftInitiated.0 = STRING: -

The above default Restart Notification Alarm Filter allows all WarmRestart, Cold2Restart, RebootRestart, and ColdRestart traps to be sent out.

4.2.2 Configuring Filters

The avCmAlmFilter MIB Group is a scratch area for all filter administration. The Objects within this MIB Group are used to configure Communication Manager, Platform/Server, and Restart filters. A snmpwalk of the avCmAlmFilter MIB Group display the OIDs defined within the MIB Group itself and their default values:

Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c private 10.129.178.85 avCmAlmFilter
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterOperation.0 = INTEGER: add(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterActive.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterResolved.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterMajor.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterMinor.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterWarning.0 = INTEGER: no(2)

AVAYA-AURA-CMALARM-MIB::avCmAlmFilterMediaGateway.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterCabinet.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterBoardNumber.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterPort.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterCategory.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterMaintenanceObject.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterExtension.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterTrunkGroup.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterTrunkMember.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterAlarmType.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterAlarmSource.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterEventID.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterWarmRestart.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterCold2Restart.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterReboot.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterCoolRestart.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterInterchange.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterCraftInitiated.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterFilterType.0 = INTEGER: cmAlarm(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterFilterIndex.0 = INTEGER: -1

4.2.2.1 Communication Manager Filters

A maximum of 240 Communication Manager Alarm filters can be added. Filters can be configured to allow:

1. All active and/or resolved Major and/or Minor and/or Warning alarms.
2. Active and/or resolved Major and/or Minor and/or Warning alarms on a Category basis.
3. Active and/or resolved Major and/or Minor and/or Warning alarms on a MO-Type basis.
4. Active and/or resolved Major and/or Minor and/or Warning alarms on an Equip-Type and location basis - avCmAlmCmFiltMediaGateway, avCmAlmCmFiltCabinet, avCmAlmCmFiltBoardNumber, avCmAlmCmFiltPort, avCmAlmCmFiltExtension, avCmAlmCmFiltTrunkGroup, and avCmAlmCmFiltTrunkMember.

A list of Maintenance Objects can be displayed by walking the avCmAlmMaintObjTable. A list of categories can be displayed by walking the avCmAlmCategoryTable. Setting both the avCmAlmCmAdjMaintenanceObject object and the avCmAlmCmAdjCategory object at the same time is not permitted. Furthermore, indexing might change when deleting existing entries.

Not all OIDS in the avCmAlmFilter MIB Group (Scratch Area) are used by Communication Manager Alarm Traps. Some OIDS are generic to all filters and some are specific to a single filter type. Here is a list of the objects that apply to Communication Manager Alarm Traps:

avCmAlmFilterOperation
avCmAlmFilterActive
avCmAlmFilterResolved
avCmAlmFilterMajor

avCmAlmFilterMinor
avCmAlmFilterWarning
avCmAlmFilterMediaGateway
avCmAlmFilterCabinet
avCmAlmFilterBoardNumber
avCmAlmFilterPort
avCmAlmFilterCategory
avCmAlmFilterMaintenanceObject
avCmAlmFilterExtension
avCmAlmFilterTrunkGroup
avCmAlmFilterTrunkMember
avCmAlmFilterFilterType
avCmAlmFilterFilterIndex

4.2.2.1.1 Adding a Communication Manager Filter

The initial default value of the avCmAlmFilterOperation object is set to add (1) and the avCmAlmFilterFilterType object is set to cmAlarm (1). Also the avCmAlmFilterFilterIndex object does not need to be set when adding a new filter. It will automatically be set to the next index for you. A Communication Manager filter must be either active, resolved or both and it must have at least one alarm severity level set. Below is an example of how to use the snmpset command to add a Resolved Major and Minor CM Alarm Filter:

```
 Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmFilterOperation.0 i 1 avCmAlmFilterActive.0 i 2
avCmAlmFilterResolved.0 i 1 avCmAlmFilterMajor.0 i 1 avCmAlmFilterMinor.0 i 1
avCmAlmFilterFilterType.0 i 1
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterOperation.0 = INTEGER: add(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterActive.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterResolved.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterMajor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterMinor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterFilterType.0 = INTEGER: cmAlarm(1)
```

After the data is successfully set in the scratch area it needs to be saved to the Filters configuration file. To do this, use the avCmAlmFilterStatus object to save it:

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmFilterStatus.0 i 4
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterStatus.0 = INTEGER: save(4)
```

Confirm the filter was added by walking the avCmAlmCmFiltTable object:

```
Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmCmFiltTable
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltActive.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltActive.1 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltResolved.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltResolved.1 = INTEGER: yes(1)
```



```
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMajor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMajor.1 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMinor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMinor.1 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltWarning.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltWarning.1 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMediaGateway.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMediaGateway.1 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltCabinet.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltCabinet.1 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltBoardNumber.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltBoardNumber.1 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltPort.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltPort.1 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltCategory.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltCategory.1 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMaintenanceObject.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMaintenanceObject.1 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltExtension.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltExtension.1 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltTrunkGroup.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltTrunkGroup.1 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltTrunkMember.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltTrunkMember.1 = INTEGER: 0
```

4.2.2.1.2 Changing a Communication Manager Filter

To change a filter you must know its index. To find out the index, perform a snmpwalk command on the avCmAlmCmFiltTable MIB Group. When changing a filter you must fill in all the objects associated with that filter, including fields you are not changing. Also, make sure you set the avCmAlmFilterOperation object to change (2), and the avCmAlmFilterFilterType object to cmAlarm (1). Below is an example of how to use the snmpset command to change the Resolved Major and Minor Communication Manager Alarm Filter added above to include warning alarms:

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmFilterOperation.0 i 2 avCmAlmFilterActive.0 i 2
avCmAlmFilterResolved.0 i 1 avCmAlmFilterMajor.0 i 1 avCmAlmFilterMinor.0 i 1
avCmAlmFilterWarning.0 i 1 avCmAlmFilterFilterIndex.0 i 1 avCmAlmFilterFilterType.0 i 1
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterOperation.0 = INTEGER: change(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterActive.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterResolved.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterMajor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterMinor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterWarning.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterFilterIndex.0 = INTEGER: 1
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterFilterType.0 = INTEGER: cmAlarm(1)
```

After the data is successfully set in the scratch area it needs to be saved to the Commutation Manager Filters configuration file. To do this, use the avCmAlmFilterStatus object to save it:


```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmFilterStatus.0 i 4
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterStatus.0 = INTEGER: save(4)
```

Confirm the filter was changed by walking the avCmAlmCmFiltTable object:

```
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterStatus.0 = INTEGER: save(4)
Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmCmFiltTable
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltActive.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltActive.1 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltResolved.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltResolved.1 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMajor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMajor.1 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMinor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMinor.1 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltWarning.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltWarning.1 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMediaGateway.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMediaGateway.1 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltCabinet.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltCabinet.1 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltBoardNumber.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltBoardNumber.1 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltPort.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltPort.1 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltCategory.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltCategory.1 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMaintenanceObject.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMaintenanceObject.1 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltExtension.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltExtension.1 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltTrunkGroup.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltTrunkGroup.1 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltTrunkMember.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltTrunkMember.1 = INTEGER: 0
```

4.2.2.1.3 Deleting a Communication Manager Filter

To delete a filter you must know its index. To find out the index perform a snmpwalk command on the avCmAlmCmFiltTable MIB Group. Once you know the index use the snmpset command to set the avCmAlmFilterOperation object to delete (3), the avCmAlmFilterFilterType object to cmAlarm (1), and the avCmAlmFilterFilterIndex object to the index you want to delete as follows:

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmFilterOperation.0 i 3 avCmAlmFilterFilterIndex.0 i 1
avCmAlmFilterFilterType.0 i 1
```


AVAYA-AURA-CMALARM-MIB::avCmAlmFilterOperation.0 = INTEGER: delete(3)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterFilterIndex.0 = INTEGER: 1
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterFilterType.0 = INTEGER: cmAlarm(1)

After the data is successfully set in the scratch area it needs to be saved to the Filters configuration file. To do this, use the avCmAlmFilterStatus object to save it:

Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c private 10.129.178.85 avCmAlmFilterStatus.0 i 4
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterStatus.0 = INTEGER: save(4)

Confirm the filter was deleted by walking the avCmAlmCmFiltTable object:

Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c private 10.129.178.85 avCmAlmCmFiltTable
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltActive.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltResolved.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMajor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMinor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltWarning.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMediaGateway.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltCabinet.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltBoardNumber.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltPort.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltCategory.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltMaintenanceObject.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltExtension.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltTrunkGroup.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmCmFiltTrunkMember.0 = INTEGER: 0

4.2.2.2 Platform/Server Filters

A maximum of 140 Platform/Server Filters can be added. Platform/Server filters can be configured to allow:

1. All active and/or resolved Major and/or Minor and/or Warning alarms
2. Active and/or resolved Major and/or Minor and/or Warning alarms on an Alarm Type basis.
3. Active and/or resolved Major and/or Minor and/or Warning alarms on a Source basis.
4. Active and/or resolved Major and/or Minor and/or Warning alarms on a Source and EventID basis.

As with Communication Manager Alarm Traps, not all OIDS in the avCmAlmFilter MIB Group (Scratch Area) are used by Platform/Server Alarm Traps. Some OIDS are generic to all filters and some are specific to a single filter type. Here is a list of the objects that apply to Platform/Server Alarm Traps:

avCmAlmFilterOperation
avCmAlmFilterActive
avCmAlmFilterResolved

avCmAlmFilterMajor
avCmAlmFilterMinor
avCmAlmFilterWarning
avCmAlmFilterAlarmType
avCmAlmFilterAlarmSource
avCmAlmFilterEventID
avCmAlmFilterFilterType
avCmAlmFilterFilterIndex
avCmAlmFilterFilterType
avCmAlmFilterFilterIndex

The list of supported Alarm Types are 1). 'A' for application alarms. 2). '*' for security alarms. 3). 'S' for system alarms. 4). 'M' for system management alarms. A list of Alarm Sources can be displayed by walking the avCmAlmPlatSrcTable. Supported EventIDs can be found in the Avaya Aura® Communication Manager Server Alarms document. Setting both the avCmAlmFilterAlarmSource object and the avCmAlmFilterAlarmType object at the same time is not permitted. The Filter indexes might change when deleting existing entries.

4.2.2.2.1 Adding a Platform/Server Filter

The configured default Platform/Server Filter permits every supported platform/server alarm. Therefore, the default filter needs be modified or deleted before adding more restrictive filters. The example below expects that the default filter is deleted. Section 4.2.2.2.3 *Deleting a Platform/Server Filter* details how to delete a Platform/Server Alarm Filter. To add a new filter set the avCmAlmFilterOperation object to add (1) and the avCmAlmFilterFilterType object to platformAlarm (2). The avCmAlmFilterFilterIndex object does not need to be set when adding a new filter. It will automatically be set to the next index for you. A filter must be either active, resolved or both and it must have at least one alarm severity level set. Below is an example of how to use the snmpset command to add an Active Major and Minor Platform/Server Alarm Filter:

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmFilterOperation.0 i 1 avCmAlmFilterActive.0 i 1
avCmAlmFilterResolved.0 i 2 avCmAlmFilterMajor.0 i 1 avCmAlmFilterMinor.0 i 1
avCmAlmFilterWarning.0 i 2 avCmAlmFilterFilterType.0 i 2
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterOperation.0 = INTEGER: add(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterActive.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterResolved.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterMajor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterMinor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterWarning.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterFilterType.0 = INTEGER:
platformAlarm(2)
```

After the data is successfully set in the scratch area it needs to be saved to the Filters configuration file. To do this, use the avCmAlmFilterStatus object to save it:

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmFilterStatus.0 i 4
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterStatus.0 = INTEGER: save(4)
```


Confirm the filter was added by walking the avCmAlmPlatFiltTable object:

```
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterFilterType.0 = INTEGER:
platformAlarm(2)
Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmPlatFiltTable
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltActive.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltResolved.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltMajor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltMinor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltWarning.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltAlarmType.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltAlarmSource.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltEventID.0 = STRING: -
```

4.2.2.2.2 Changing a Platform/Server Filter

To change a Platform/Server filter you must know its index. To find out the index perform a snmpwalk command on the avCmAlmPlatFiltTable MIB Group. When changing a filter you must fill in all the objects associated with that filter, including fields you are not changing. Also, make sure you set the avCmAlmFilterOperation object to change (2), and the avCmAlmFilterFilterType object to platformAlarm (2). Below is an example of how to use the snmpset command to change the Active Major and Minor Platform/Server Alarm Filter added above to include warning and resolved alarms:

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c       -c
private 10.129.178.85  avCmAlmFilterOperation.0 i 2 avCmAlmFilterActive.0 i 1 avCmAlmFil
terResolved.0 i 1 avCmAlmFilterMajor.0 i 1 avCmAlmFilterMinor.0 i 1 avCmAlmFilterWarni
ng.0 i 1 avCmAlmFilterFilterIndex.0 i 0 avCmAlmFilterFilterType.0 i 2
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterOperation.0 = INTEGER: change(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterActive.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterResolved.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterMajor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterMinor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterWarning.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterFilterIndex.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterFilterType.0 = INTEGER:
platformAlarm(2)
```

After the data is successfully set in the scratch area it needs to be saved to the Filters configuration file. To do this, use the avCmAlmFilterStatus object to save it:

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmFilterStatus.0 i 4
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterStatus.0 = INTEGER: save(4)
```

Confirm the filter was changed by walking the avCmAlmPlatFiltTable object:


```
Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmPlatFiltTable
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltActive.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltResolved.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltMajor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltMinor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltWarning.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltAlarmType.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltAlarmSource.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltEventID.0 = STRING: -
```

4.2.2.2.3 Deleting a Platform/Server Filter

To delete a filter you must know its index. To find out the index perform a snmpwalk command on the avCmAlmPlatFiltTable MIB Group.

```
Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmPlatFiltTable
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltActive.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltResolved.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltMajor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltMinor.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltWarning.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltAlarmType.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltAlarmSource.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltEventID.0 = STRING: -
```

Once you know the index use the snmpset command to set the avCmAlmFilterOperation object to delete (3), the avCmAlmFilterFilterType object to platformAlarm (2), and the avCmAlmFilterFilterIndex object to the index you want to delete.

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmFilterOperation.0 i 3 avCmAlmFilterFilterIndex.0 i 0
avCmAlmFilterFilterType.0 i 2
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterOperation.0 = INTEGER: delete(3)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterFilterIndex.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterFilterType.0 = INTEGER:
platformAlarm(2)
```

After the data is successfully set in the scratch area it needs to be saved to the Filters configuration file. To do this, use the avCmAlmFilterStatus object to save it:

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmFilterStatus.0 i 4
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterStatus.0 = INTEGER: save(4)
```

Confirm the filter was deleted by walking the avCmAlmCmFiltTable object:

```
Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmPlatFiltTable
```


AVAYA-AURA-CMALARM-MIB::avCmAlmPlatFiltTable = No Such Object available on this agent at this OID

4.2.2.3 Restart Notification Filters

A maximum of 5 Restart Notification Filters can be configured. As with Communication Manager and Platform/Server Alarm Traps, not all OIDS in the avCmAlmFilter MIB Group (Scratch Area) are used by Restart Notification Alarm Traps. Some OIDS are generic to all filters and some are specific to a single filter type. Here is a list of the objects that apply to Restart Alarm Traps:

avCmAlmFilterOperation
avCmAlmFilterWarmRestart
avCmAlmFilterCold2Restart
avCmAlmFilterReboot
avCmAlmFilterCoolRestart
avCmAlmFilterInterchange
avCmAlmFilterCraftInitiated
avCmAlmFilterFilterType
avCmAlmFilterFilterIndex

4.2.2.3.1 Adding a Restart Notification Filter

The configured default Restart Notification Filter allows every supported restart alarm. Therefore, the default filter needs be modified or deleted before adding more restrictive filters. The example below expects that the default filter is deleted. To delete a Restart Notification Alarm Filter see section *4.2.2.3.3 Deleting a Restart Notification Filter*. To add a new filter set the avCmAlmFilterOperation object to add (1) and the avCmAlmFilterFilterType object to restartNotification (3). The avCmAlmFilterFilterIndex object does not need to be set when adding a new filter. It will automatically be set to the next index for you. A valid restart filter need only contain one restart level. Below is an example of how to use the snmpset command to add an avCmAlmRstFiltWarmRestart, avCmAlmRstFiltCold2Restart and, avCmAlmRstFiltRebootRestart Alarm Notification Filter.

Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c private 10.129.178.85 avCmAlmFilterOperation.0 i 1 avCmAlmFilterWarmRestart.0 i 1 avCmAlmFilterCold2Restart.0 i 1 avCmAlmFilterReboot.0 i 1 avCmAlmFilterFilterType.0 i 3 avCmAlmFilterCoolRestart.0 i 2 avCmAlmFilterInterchange.0 s "-" avCmAlmFilterCraftInitiated.0 s "-"
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterOperation.0 = INTEGER: add(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterWarmRestart.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterCold2Restart.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterReboot.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterFilterType.0 = INTEGER: restartNotification(3)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterCoolRestart.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterInterchange.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterCraftInitiated.0 = STRING: -

After the data is successfully set in the scratch area it needs to be saved to the Filters configuration file. To do this, use the avCmAlmFilterStatus object to save it:

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmFilterStatus.0 i 4
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterStatus.0 = INTEGER: save(4)
```

Confirm the filter was deleted by walking the avCmAlmRstFiltTable object:

```
Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmRstFiltTable
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltWarmRestart.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltCold2Restart.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltReboot.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltCoolRestart.0 = INTEGER: no(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltInterchange.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltCraftInitiated.0 = STRING: -
```

4.2.2.3.2 Changing a Restart Notification Filter

To change a filter you must know its index. To find out the index perform a snmpwalk command on the avCmAlmRstFiltTable MIB Group. When changing a filter you must fill in all the objects associated with that filter, including fields you are not changing. Also, make sure you set the avCmAlmFilterOperation object to change (2), and the avCmAlmFilterFilterType object to restartNotification (3). Below is an example of how to use the snmpset command to change the avCmAlmRstFiltWarmRestart, avCmAlmRstFiltCold2Restart and, avCmAlmRstFiltRebootRestart added above to include avCmAlmFilterInterchange notification alarms:

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmFilterOperation.0 i 2 avCmAlmFilterWarmRestart.0 i 1
avCmAlmFilterCold2Restart.0 i 1 avCmAlmFilterReboot.0 i 1 avCmAlmFilterFilterType.0 i 3
avCmAlmFilterCoolRestart.0 i 1 avCmAlmFilterInterchange.0 s "-"
avCmAlmFilterCraftInitiated.0 s "-"  avCmAlmFilterFilterIndex.0 i 0
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterOperation.0 = INTEGER: change(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterWarmRestart.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterCold2Restart.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterReboot.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterFilterType.0 = INTEGER:
restartNotification(3)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterCoolRestart.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterInterchange.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterCraftInitiated.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterFilterIndex.0 = INTEGER: 0
```

After the data is successfully set in the scratch area it needs to be saved to the Filters configuration file. To do this, use the avCmAlmFilterStatus object to save it:

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmFilterStatus.0 i 4
```



```
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterStatus.0 = INTEGER: save(4)
```

Confirm the filter was changed by walking the avCmAlmRstFiltTable object:

```
Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmRstFiltTable
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltWarmRestart.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltCold2Restart.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltReboot.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltCoolRestart.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltInterchange.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltCraftInitiated.0 = STRING: -
```

4.2.2.3.3 Deleting a Restart Filter

To delete a filter you must know its index. Use the snmpwalk command to walk the avCmAlmRstFiltTable to find the index.

```
Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmRstFiltTable
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltWarmRestart.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltCold2Restart.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltReboot.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltCoolRestart.0 = INTEGER: yes(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltInterchange.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltCraftInitiated.0 = STRING: -
```

Once you know the index use the snmpset command to set the avCmAlmFilterOperation to delete (3), the avCmAlmFilterFilterType to restartNotification (3), and the avCmAlmFilterFilterIndex to the index you want to delete.

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmFilterOperation.0 i 3 avCmAlmFilterFilterIndex.0 i 0
avCmAlmFilterFilterType.0 i 3
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterOperation.0 = INTEGER: delete(3)
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterFilterIndex.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterFilterType.0 = INTEGER:
restartNotification(3)
```

After the data is successfully set in the scratch area it needs to be saved to the Filters configuration file. To do this, use the avCmAlmFilterStatus object to save it:

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmFilterStatus.0 i 4
AVAYA-AURA-CMALARM-MIB::avCmAlmFilterStatus.0 = INTEGER: save(4)
```

Confirm the filter was deleted by walking the avCmAlmCmFiltTable object:


```
Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmRstFiltTable
AVAYA-AURA-CMALARM-MIB::avCmAlmRstFiltTable = No Such Object available on this
agent at this OID
```

4.3 Alarm Level Adjustment Feature

The Alarm Level Adjustment Feature is new in CM Release 6.3.1xx and CM Release7.0. It allows users to change the severity of a Communication Manager Alarm Trap or a Platform Alarm Trap. In addition to changing an alarm's severity level the feature also allows the alarm to be discarded. Alarms are processed by the Alarm Level Adjustment feature before being processed by the Alarm Filters Feature. An alarms severity can only be displayed or changed using SNMP get or set commands. There is no Alarm Level Adjustment SMI Page. NOTE: The alarm level change only applies to FP Traps. It does not change INADS traps which go to Avaya Services.

4.3.1 Default Settings:

None.

4.3.2 Configuring the Alarm Level Adjustment Feature

The avCmAlmAdjust MIB Group is a scratch area for all alarm adjustment administration. The objects within the MIB Group are used to configure Communication Manager and Platform entries. A snmpwalk of the avCmAlmAdjust MIB Group displays the OIDs defined within the MIB Group itself and their default values:

```
Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmAdjust
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustOperation.0 = INTEGER: add(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustMaintenanceObject.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustCategory.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustMajorOnBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustMajorOffBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustMinorOnBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustMinorOffBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustWarningOnBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustWarningOffBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustAlarmSource.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustEventID.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustServAlarmType.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustServMajor.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustServMinor.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustServWarning.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustAlarmType.0 = INTEGER: cmAlarm(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustAlarmIndex.0 = INTEGER: -1
```

4.3.2.1 Communication Manager Alarm Level Adjustment Entries

A maximum of 200 Communication Manager Alarm Level Adjustment entries are allowed to be configured. Alarm Severity levels can be changed for:

1. Major, minor, and warning on/off board alarms for individual Maintenance Objects (MOs).
2. Major, minor, and warning on/off board alarms for alarm categories.
3. All major, minor, and warning on/off board alarms.

Communication Manger Alarm adjustment entries are parsed from the top of the list down. Major on/off board alarms can be set to blank (1), discard (2), minor (4), and warning (5). Minor on/off board alarms can be set to blank (1), discard (2), major (3), warning (5), and original (6). Warning on/off board alarms can be set to blank (1), discard (2), major (3), minor (4), and original (6). If you want to increase or decrease an alarms severity level set the major, minor, and warning attributes. If you want to discard an alarm, use the discard attribute. If the alarm has been modified by the System Access Terminal (SAT) ***set options*** command/form on CM, the original (6) attribute can be used to set it back to the original alarm level. A list of Maintenance Objects can be displayed by walking the avCmAlmMaintObjTable. A list of categories can be displayed by walking the avCmAlmCategoryTable. Setting both the avCmAlmCmAdjMaintenanceObject object and the avCmAlmCmAdjCategory object at the same time is not permitted. Furthermore, indexing might change when adding, changing, and deleting existing entries.

Not all OIDS in the avCmAlmAdjust MIB Group (Scratch Area) are used by Communication Manager Alarm Traps. Some OIDS are generic to all filters and some are specific to a single filter type. Here is a list of the objects that apply to Communication Manager Alarms:

avCmAlmAdjustOperation
avCmAlmAdjustMaintenanceObject
avCmAlmAdjustCategory
avCmAlmAdjustMajorOnBrd
avCmAlmAdjustMajorOffBrd
avCmAlmAdjustMinorOnBrd
avCmAlmAdjustMinorOffBrd
avCmAlmAdjustWarningOnBrd
avCmAlmAdjustWarningOffBrd
avCmAlmAdjustAlarmType
avCmAlmAdjustAlarmIndex

4.3.2.1.1 Adding a Communication Manager Alarm Level Adjustment Entry

The initial default settings for the avCmAlmAdjustOperation object is set to add (1) and the avCmAlmAdjustAlarmType object is set to cmAlarm (1). Also the avCmAlmAdjustAlarmIndex object does not need to be set when adding a new filter. It will automatically be set to the next index for you. An alarm entry must have at least one severity level. Below is an example of how to use the snmpset command to add an entry that will change the alarm severity of all off board trunk warning alarms to minor:

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmAdjustOperation.0 i 1 avCmAlmAdjustMajorOffBrd.0 i 1
```



```
avCmAlmAdjustMajorOnBrd.0 i 1 avCmAlmAdjustMinorOffBrd.0 i 1
avCmAlmAdjustMinorOnBrd.0 i 1 avCmAlmAdjustWarningOnBrd.0 i 1
avCmAlmAdjustWarningOffBrd.0 i 4 avCmAlmAdjustCategory.0 s "trunks"
avCmAlmAdjustAlarmType.0 i 1
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustOperation.0 = INTEGER: add(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustMajorOffBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustMajorOnBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustMinorOffBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustMinorOnBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustWarningOnBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustWarningOffBrd.0 = INTEGER: minor(4)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustCategory.0 = STRING: trunks
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustAlarmType.0 = INTEGER: cmAlarm(1)
```

After the data is successfully set in the scratch area it needs to be saved to the Adjustment configuration file. To do this, use the avCmAlmAdjustStatus object to save it:

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmAdjustStatus.0 i 4
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustStatus.0 = INTEGER: save(4)
```

Confirm the entry was added by walking the avCmAlmCmAdjTable object:

```
Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmCmAdjTable
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjMaintenanceObject.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjCategory.0 = STRING: trunks
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjMajorOnBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjMajorOffBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjMinorOnBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjMinorOffBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjWarningOnBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjWarningOffBrd.0 = INTEGER: minor(4)
```

4.3.2.2 Changing a Communication Manager Alarm Level Adjustment Entry

To change an Alarm Level Adjustment entry you must know its index. To find out the index perform a snmpwalk command on the avCmAlmCmAdjTable MIB Group. When changing an alarm entry you must fill in all the objects associated with that alarm, including fields you are not changing. Also, make sure you set the avCmAlmAdjustOperation object to change (2), and the avCmAlmAlarmType object to cmAlarm (1). Below is an example of how to use the snmpset command to change the Alarm Level Adjustment entry added above to include upgrading on board warning alarms to minor alarms:

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmAdjustOperation.0 i 2 avCmAlmAdjustMajorOffBrd.0 i 1
avCmAlmAdjustMajorOnBrd.0 i 1 avCmAlmAdjustMinorOffBrd.0 i 1
avCmAlmAdjustMinorOnBrd.0 i 1 avCmAlmAdjustWarningOnBrd.0 i 4
avCmAlmAdjustWarningOffBrd.0 i 4 avCmAlmAdjustCategory.0 s "trunks"
avCmAlmAdjustAlarmType.0 i 1 avCmAlmAdjustAlarmIndex.0 i 0
```



```
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustOperation.0 = INTEGER: change(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustMajorOffBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustMajorOnBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustMinorOffBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustMinorOnBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustWarningOnBrd.0 = INTEGER: minor(4)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustWarningOffBrd.0 = INTEGER: minor(4)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustCategory.0 = STRING: trunks
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustAlarmType.0 = INTEGER: cmAlarm(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustAlarmIndex.0 = INTEGER: 0
```

After the data is successfully set in the scratch area it needs to be saved to the Filters configuration file. To do this, use the avCmAlmAdjustStatus object to save it:

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85 avCmAlmAdjustStatus.0 i 4
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustStatus.0 = INTEGER: save(4)
```

Confirm the filter was changed by walking the avCmAlmCmAdjTable object:

```
Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmCmAdjTable
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjMaintenanceObject.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjCategory.0 = STRING: trunks
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjMajorOnBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjMajorOffBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjMinorOnBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjMinorOffBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjWarningOnBrd.0 = INTEGER: minor(4)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjWarningOffBrd.0 = INTEGER: minor(4)
```

4.3.2.3 Deleting a Communication Manager Alarm Level Adjustment Entry

To delete an Alarm Level Adjustment entry you must know its index. Use the snmpwalk command to walk the avCmAlmCmAdjTable to find the index.

```
Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmCmAdjTable
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjMaintenanceObject.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjCategory.0 = STRING: trunks
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjMajorOnBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjMajorOffBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjMinorOnBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjMinorOffBrd.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjWarningOnBrd.0 = INTEGER: minor(4)
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjWarningOffBrd.0 = INTEGER: minor(4)
```

Once you know the index, use the snmpset command to set the avCmAlmAdjustOperation to delete (3), the avCmAlmAlarmType to cmAlarm(1) (if it is not already set) and the avCmAlmAdjustAlarmIndex to the index you want to delete.


```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmAdjustOperation.0 i 3  avCmAlmAdjustAlarmIndex.0 i 0
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustOperation.0 = INTEGER: delete(3)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustAlarmIndex.0 = INTEGER: 0
```

After the data is successfully set in the scratch area it needs to be saved to the Adjustments configuration file. To do this, use the avCmAlmAdjustStatus object to save it:

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmAdjustStatus.0 i 4
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustStatus.0 = INTEGER: save(4)
```

Confirm the alarm entry was deleted by walking the avCmAlmCmAdjTable object:

```
Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmCmAdjTable
AVAYA-AURA-CMALARM-MIB::avCmAlmCmAdjTable = No Such Object available on this
agent at this OID
```

4.3.3 Platform/ Server Alarm Level Adjustment Entries

A maximum of 80 Server/Platform Alarm Level Adjustment entries are allowed to be configured. Server/Platform Alarm Severity Levels can be changed for:

1. Major, minor, and warning alarms for different Sources and EventIDs
2. Major, minor, and warning alarms for different Sources
3. Major, minor, and warning alarms for different alarm Types.
4. All major, minor, and warning alarms.

Platform adjustment entries are parsed from the top of the list down. Major alarms can be set to blank (1), discard (2), minor (4), and warning (5). Minor on/off board alarms can be set to blank (1), discard (2), major (3), warning (5), and original (6). Warning alarms can be set to blank (1), discard (2), major (3), minor (4), and original (6). If you want to increase or decrease an alarms severity level set the major, minor, and warning attributes. If you want to discard an alarm, use the discard attribute. Setting both the avCmAlmAdjustAlarmSource object and the avCmAlmAdjustServAlarmType object at the same time is not permitted. The list of supported Alarm Types is: 1). 'A' for application alarms. 2). '*' for security alarms. 3). 'S' for system alarms. 4). 'M' for system management alarms. A list of Alarm Sources can be displayed by walking the avCmAlmPlatSrcTable. Supported EventIDs can be found in the Avaya Aura ® Communication Manager Server Alarms document. Furthermore, indexing might change when adding, changing, and deleting existing entries.

Not all OIDS in the avCmAlmAdjust MIB Group (Scratch Area) are used by Server/Platform Alarm Traps. Some OIDS are generic to all alarms and some are specific to Server/Platform Alarms. Here is a list of the objects that apply to Platform Alarm Traps:

avCmAlmAdjustOperation
avCmAlmAdjustAlarmSource

avCmAlmAdjustEventID
avCmAlmAdjustServAlarmType
avCmAlmAdjustServMajor
avCmAlmAdjustServMinor
avCmAlmAdjustServWarning
avCmAlmAdjustAlarmType
avCmAlmAdjustAlarmIndex

4.3.3.1 Adding a Server/Platform Alarm Level Adjustment Entry

By default the avCmAlmAdjustOperation object is set to add (1) and the avCmAlmAdjustAlarmType object is set to cmAlarm (1). Therefore the avCmAlmAdjustAlarmType object needs to be set to serverAlarm(2). Also the avCmAlmAdjustAlarmIndex object does not need to be set when adding a new entry. It will automatically be set to the next index for you. An alarm entry must have at least one severity level set. Below is an example of how to use the snmpset command to add an entry that will change the alarm severity of all minor SVC_MON alarms to a warning:

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmAdjustOperation.0 i 1 avCmAlmAdjustServMajor.0 i 1
avCmAlmAdjustServMinor.0 i 5 avCmAlmAdjustServWarning.0 i 1
avCmAlmAdjustAlarmSource.0 s "SVC_MON" avCmAlmAdjustAlarmType.0 i 2
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustOperation.0 = INTEGER: add(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustServMajor.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustServMinor.0 = INTEGER: warning(5)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustServWarning.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustAlarmSource.0 = STRING: SVC_MON
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustAlarmType.0 = INTEGER:
serverAlarm(2)
```

After the data is successfully set in the scratch area it needs to be saved to the Adjustment configuration file. To do this, use the avCmAlmAdjustStatus object to save it:

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmAdjustStatus.0 i 4
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustStatus.0 = INTEGER: save(4)
```

Confirm the entry was added by walking the avCmAlmCmAdjTable object:

```
Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmPlatAdjTable
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjAlarmSource.0 = STRING: SVC_MON
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjEventID.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjAlarmType.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjMajor.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjMinor.0 = INTEGER: warning(5)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjWarning.0 = INTEGER: blank(1)
```

4.3.2.2 Changing a Platform/Server Alarm Level Adjustment Entry

To change an Alarm Level Adjustment entry you must know its index. To find out the index perform a snmpwalk command on the avCmAlmPlatAdjTable MIB Group. When changing an alarm entry you must fill in all the objects associated with that alarm, including fields you are not changing. Also, make sure you set the avCmAlmAdjustOperation object to change (2), and the avCmAlmAlarmType object to serverAlarm (2). Below is an example of how to use the snmpset command to change the Alarm Level Adjustment entry added above to include checking for a specific eventID:

Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c private 10.129.178.85 avCmAlmAdjustOperation.0 i 2 avCmAlmAdjustServMajor.0 i 1 avCmAlmAdjustServMinor.0 i 5 avCmAlmAdjustServWarning.0 i 1 avCmAlmAdjustAlarmSource.0 s "SVC_MON" avCmAlmAdjustEventID.0 s "2" avCmAlmAdjustAlarmIndex.0 i 0 avCmAlmAdjustAlarmType.0 i 2
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustOperation.0 = INTEGER: change(2)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustServMajor.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustServMinor.0 = INTEGER: warning(5)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustServWarning.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustAlarmSource.0 = STRING: SVC_MON
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustEventID.0 = STRING: 2
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustAlarmIndex.0 = INTEGER: 0
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustAlarmType.0 = INTEGER: serverAlarm(2)

After the data is successfully set in the scratch area it needs to be saved to the Filters configuration file. To do this, use the avCmAlmAdjustStatus object to save it:

Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c private 10.129.178.85 avCmAlmAdjustStatus.0 i 4
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustStatus.0 = INTEGER: save(4)

Confirm the filter was changed by walking the avCmAlmPlatAdjTable object:

AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustStatus.0 = INTEGER: save(4)
Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c private 10.129.178.85 avCmAlmPlatAdjTable
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjAlarmSource.0 = STRING: SVC_MON
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjEventID.0 = STRING: 2
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjAlarmType.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjMajor.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjMinor.0 = INTEGER: warning(5)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjWarning.0 = INTEGER: blank(1)

4.3.3.3 Deleting a Platform/Server Alarm Level Adjustment Entry

To delete a Platform/Server alarm entry you must know its index. Use the snmpwalk command to walk the avCmAlmPlatAdjTable to find the index.


```
Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmPlatAdjTable
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjAlarmSource.0 = STRING: SVC_MON
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjAlarmSource.1 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjEventID.0 = STRING: 2
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjEventID.1 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjAlarmType.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjAlarmType.1 = STRING: A
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjMajor.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjMajor.1 = INTEGER: warning(5)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjMinor.0 = INTEGER: warning(5)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjMinor.1 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjWarning.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjWarning.1 = INTEGER: blank(1)
```

Once you know the index use the snmpset command to set the avCmAlmAdjustOperation to delete (3), the avCmAlmAlarmType to serverAlarm(1) if it is not already set, and the avCmAlmAdjustAlarmIndex to the index you want to delete.

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmAdjustOperation.0 i 3  avCmAlmAdjustAlarmIndex.0 i 0
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustOperation.0 = INTEGER: delete(3)
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustAlarmIndex.1 = INTEGER: 0
```

After the data is successfully set in the scratch area it needs to be saved to the Adjustments configuration file. To do this, use the avCmAlmAdjustStatus object to save it:

```
Server1> snmpset -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmAdjustStatus.0 i 4
AVAYA-AURA-CMALARM-MIB::avCmAlmAdjustStatus.0 = INTEGER: save(4)
```

Confirm the alarm entry was deleted by walking the avCmAlmCmAdjTable object:

```
Server1> snmpwalk -m /usr/share/snmp/mibs/AVAYA-AURA-CMALARM-MIB.txt -v2c -c
private 10.129.178.85  avCmAlmPlatAdjTable
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjAlarmSource.0 = STRING: SVC_MON
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjEventID.0 = STRING: 2
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjAlarmType.0 = STRING: -
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjMajor.0 = INTEGER: blank(1)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjMinor.0 = INTEGER: warning(5)
AVAYA-AURA-CMALARM-MIB::avCmAlmPlatAdjWarning.0 = INTEGER: blank(1)
```

5.0 Load Agent.

The LoadAgent was replaced by the CMLoadAgent. The Load Agent is used to download firmware to CM circuit packs/boards that support the firmware download feature. The functionality of the Load Agent was not changed. The only minor change to the Load Agent is that it no longer displays a system's entire circuit pack inventory.

The New CMLoadAgent has been modified to only display downloadable boards. Other than this change, users should not notice any difference.

6.0 SNMP CLI commands

The CM SNMP CLI commands can be used to configure SNMP user access, SNMP FP Trap destinations and incoming trap authorization checking. Prior to CM Release 6.3.1xx only users with root permission could access the SNMP CLI commands. However, starting in CM Release 6.3.1xx any user can access them. The following table outlines the changes in the commands from CM Release 6.3.xx to CM Release 6.3.1xx

Old SNMP Command	New SNMP Command	Description of the Changes
snmpaccessconfig		This command was removed. The **–i** option was added to the snmpuserconfig command.
snmpuserconfig	**snmpuserconfig**	• The –i option was added to input a valid IP Address or 0.0.0.0 for ***Any Valid IP Address***. • The –D option was removed.
snmptrapconfig	**snmptrapconfig**	• The –d option was changed to –i • The –i option was changed to –inform • The –t option was changed to –trap • The –D option was removed • The –old_dest was changed to –old_ipaddr • The –p options was added to support changing the default port from port 162.
snmpagentsstatus	**snmpagentsstatus**	• The MIB-2 SubAgent is not displayed. It is co-resident with the MasterAgent

		• The CM SNMP SubAgents names have changed.
testcustalm	**testcustalm**	No changes
	snmpconfig	New Command to retrieve SNMP Engine ID and to reset the snmpd.conf and snmptrapd.conf configuration files to original default settings.
	snmpinctrapconfig	New Command to configure incoming trap authorization checking for the snmptrapd

The existing CM SNMP CLI commands were modified to support configuring the system using Net-SNMP directives. All three CM SNMP administration commands configure SNMP users using USM and VACM. Unlike the previous implementation, the MasterAgent no longer has to be manually stopped and started after making any administrative changes. Internally a random SNMPv3 user is added whenever any type of modification or query is required. Once its task is completed it is removed.

6.1 snmpuserconfig command

The *snmpuserconfig* command allows users to configure SNMPv1, SNMPv2c, and SNMPv3 access to a system. Up to 25 access entries are supported. The old *snmpuserconfig* command was modified to support Net-SNMP administration. The createUser directive is used to create SNMPv3 users so that the authentication and privacy passwords are stored encrypted. The minimum security model supported by the *snmpuserconfig* command is authNoPriv. CM does not support a security model of noAuthNoPriv. Usage for the *snmpuserconfig* command is as follows:

```
server1> snmpuserconfig --help
usage:
  snmpuserconfig -v 1|2  -r|-w -i <ipaddr> [--old_ipaddr <ipaddr>]
              -c <comm_name> [-o <comm_name>]
              --add | --delete | --change
  snmpuserconfig -v 3  -r|-w -u <user> [-o <user>] [-s authNoPriv | authPriv]
              [-a <auth_passwd>] [--auth_prot <protocol>]
              [-p <priv_passwd>] [--priv_prot <protocol>]
```



```
              --add | --delete | --change
 snmpuserconfig --help | -h
  whereas:
   -v          : SNMP version (1, 2, 3)
   -r          : read-only entry
   -i          : ip address of polling system (v1 and v2c only)
   --old_ipaddr : previous IP address of polling system
                 (only valid with the 'change' option,
                  required when changing the IP Address for an entry).
   -w          : read-write entry
   -c          : v1/v2c community name
   -u          : v3 user name
   -o          : previous community/user name
                 (only valid with the 'change' option,
                  required when changing the user/community name).
   -s          : v3 security model (authNoPriv, authPriv)
   -a          : v3 authentication password
   --auth_prot  : v3 authentication protocol (MD5, SHA)
   -p          : v3 privacy password
   --priv_prot  : v3 privacy protocol (none, DES, AES128)
   --add        : add entry
   --change     : change entry
   --delete     : delete entry
   --auth_list  : list supported authentication protocols.
   --priv_list  : list supported privacy protocols.
   --rclist     : list possible return codes from this command.
   --rccode     : obtain error string associated with a return code.
   --help|-h    : usage (this)

DEFAULT: 'snmpuserconfig' called with no arguments will display the current
         configuration.

NOTE: Using the -a and -p option with arguments on the command line presents a
      security risk as the passwords may be logged. Use the -a and -p options
      without an argument to prevent this.
```

6.1.1 SNMPv1/v2c

SNMPv1 and SNMPv2c access entries can be configured using the *snmpuserconfig* command. The following options apply to SNMPv1/v2c administration:

-v	Version 1 or 2c
-c	Community Name - A string with a minimum length of 1 and a max length of 31. It can contain alphanumeric characters. However certain charters such as ‘`’ or ‘\’’or ‘\\’ or ‘&’ or ‘,’ or ‘ ‘ or ‘=’ or ‘;’ or ‘”’
-r/-w	Read or Read-Write
-i	A valid IP Address or 0.0.0.0
-o	Old Community Name (change command only)
–old_ipaddr	Old IP Address (change command only)

--add	Add an entry.
--change	Change an entry.
--delete	Delete an entry

The –i option is used to either add, change, or delete an *Any Valid IP Address* (0.0.0.0) entry or an *Individual IP address* entry such as 10.129.178.85. Administration of an *Any Valid IP Address* entry allows any IP address with the correct Community Name to access the system. Administration of an *Individual IP Address* entry only allows access by that IP Address as long as it has the correct Community String. SNMPv1/v2c entries to be added, changed, and deleted using the *snmpuserconfig* command as follows:

The following command is used to add a SNMPv1 *Any Valid IP Address* entry:

snmpuserconfig –v 1 –c public –r –i 0.0.0.0 –add

The following command is used to add a SNMPv2 *Individual IP Address* entry:

snmpuserconfig –v 2c –c private –w –i 10.129.175.80 –add

Only a SNMPv1/v2c entries IP Address and the Community Name can be changed. The following command is used to change a SNMPv1 *Any Valid IP Address* entry:

snmpuserconfig –v 1 –c notpublic -o public –r –i 10.129.178.80 –-old_ipaddr 0.0.0.0 –change

The following command is used to add a SNMPv2 *Individual IP Address* entry to an *Any Valid IP Address* entry:

snmpuserconfig –v 2c –c private –w –i 0.0.0.0 –old_ipaddr 10.129.175.80 –change

The following command is used to delete a SNMPv1 Individual IP Address entry:

snmpuserconfig –v 1 –c notpublic –r –i 10.129.175.80 –del

The following command is used to delete a SNMPv2 Any Valid IP Address entry:

snmpuserconfig –v 2c –c private –w –i 0.0.0.0 –del

6.1.2 SNMPv3

SNMPv3 entries can be configured using the *snmpuserconfig* command with the following options

-v	Version 3

-u	User Name - A string with a minimum length of 1 and a max length of 31. It can contain alphanumeric characters. However certain charters such as ‘`’ or ‘\’’or ‘\\’ or ‘&’ or ‘,’ or ‘ ‘ or ‘=’ or ‘;’ or ‘”’
-r/-w	Read or Read-Write
-s	Security Model – CM supports authNoPriv and authPriv
-a	Authentication Password – An alphanumeric string with a minimum length of 8 and a max length of 64. The password cannot contain certain characters such as ‘`’ or ‘\’’or ‘\\’ or ‘&’ or ‘,’ or ‘ ‘or ‘”’ or ‘-‘
--auth_prot	Authentication Protocol MD5 or SHA.
-p	Privacy Password – An alphanumeric string with a minimum length of 8 and a max length of 64. The password cannot contain certain characters such as ‘`’ or ‘\’’or ‘\\’ or ‘&’ or ‘,’ or ‘ ‘or ‘”’ or ‘-‘
--priv_prot	Privacy Protocol of DES or AES128
-o	Old User Name (change command only)
--add	Add an entry.
--change	Change an entry.
--delete	Delete an entry

SNMPv3 entries to be added, changed, and deleted using the snmpuserconfig command as follows:

The following command is used to add a SNMPv3 entry:

snmpuserconfig –v 3 –u username –r -s authPriv –a authenitication pw –auth_prot MD5 –p privacypw –priv_prot DES –add

A SNMPv3 user name an passwords can be changed. The following command is used to change a SNMPv3 entries user name:

snmpuserconfig –v 3 –u newusername –o username –r -s authPriv –a authenitication pw –auth_prot MD5 –p privacypw –priv_prot DES –change

The following command is used to change a SNMPv3 authentication and privacy passwords:

snmpuserconfig –v 3 –u newusername –r -s authPriv –a newauthenitication pw –auth_prot MD5 –p newprivacypw –priv_prot DES –change

The following command is used to delete a SNMPv3 entry:

snmpuserconfig –v 3 –u newusername –r -s authPriv –a newauthenitication pw –auth_prot MD5 –p newprivacypw –priv_prot DES –del

6.2 snmptrapconfig command

The *snmptrapconfig* command allows users to configure SNMPv1, SNMPv2c, and SNMPv3 trap destinations to a system. Up to 25 trap destination entries are supported. The existing snmptrapconfig command was modified to support Net-SNMP administration. Additionally a **–p** option was added to support changing the outgoing port from the default value of 162. The createUser directive is used to create SNMPv3 users so that the authentication and privacy passwords are stored encrypted. The minimum security model supported by the *snmptraprconfig* command is authNoPriv. CM does not support a security model of noAuthNoPriv. Usage for the *snmptraprconfig* command is as follows:

```
server1> snmptrapconfig --help
usage:
 snmptrapconfig -v 1|2  --inform|--trap  -i <ipaddr> [--old_ipaddr <ipaddr>]
             --port <number> -c <comm_name> [-o <comm_name>]
             --add | --delete | --change
 snmptrapconfig -v 3  --inform|--trap  -i <ipaddr> [--old_ipaddr <ipaddr>]
             -u <user> [-o <user>]  [-s authNoPriv | authPriv]
             [-a <auth_passwd>] [--auth_prot <protocol>]
             [-p <priv_passwd>] [--priv_prot <protocol>]
             --add | --delete | --change
 snmptrapconfig --help | -h
  whereas:
  -i          : destination IP address
  --old_ipaddr : previous destination IP address
               (only valid with the 'change' option,
                required when changing the destination for an entry).
  -v          : SNMP version (1, 2, 3)
  --inform     : inform entry (v2 & v3 only)
  --trap       : trap entry
  --old_type   : previous entry type (inform / trap)
               (only valid with the 'change' option,
                required when changing the type for an entry).
  --port       : Port number (default is 162)
  -c          : v1/v2c community name
  -u          : v3 user name
  -o          : previous community/user name
               (only valid with the 'change' option,
                required when changing the user/community name).
  -s          : v3 security model (authNoPriv, authPriv)
  -a          : v3 authentication password
  --auth_prot  : v3 authentication protocol (MD5, SHA)
  -p          : v3 privacy password
  --priv_prot  : v3 privacy protocol (none, DES, AES128)
  -e          : Engine ID (used with -i option)
  --add        : add entry
  --change     : change entry
  --delete     : delete entry
  --auth_list  : list supported authentication protocols.
  --priv_list  : list supported privacy protocols.
  --rclist     : list possible return codes from this command.
```


--rccode : obtain error string associated with a return code.
--help|-h : usage (this)

DEFAULT: 'snmptrapconfig' called with no arguments will display the current configuration.

NOTE: Using the -a and -p option with arguments on the command line presents a security risk as the passwords may be logged. Use the -a and -p options without an argument to prevent this.

6.2.1 SNMPv1/v2c

SNMPv1 and SNMPv2c entries can be configured using the *snmptrapconfig c*ommand with the following options:

-v	Version 1 or 2c
-c	Community Name - A string with a minimum length of 1 and a max length of 31. It can contain alphanumeric characters. However certain charters such as ‘`’ or ‘\’’or ‘\\’ or ‘&’ or ‘,’ or ‘ ‘ or ‘=’ or ‘;’ or ‘”’
--trap/--inform	Trap type – Trap (SNMPv1 and SNMPv2v) or inform (SNMPv2c)
-i	A valid IP Address
--port	Port - Allows users to change the destination port (default is 162)
-o	Old Community Name (change command only)
–old_ipaddr	Old IP Address (change command only)
--old_type	Old Type (change command only)
--add	Add an entry.
--change	Change an entry.
--delete	Delete an entry

SNMPv1/v2c entries to be added, changed, and deleted using the *snmptrapconfig* command as follows:

The following command is used to add a SNMPv1 trap destination:

snmptrapconfig –v 1 –c public –-trap –i 10.129.178.79 –port 10162 –add

The following command is used to add a SNMPv2c inform destination:

snmptrapconfig –v 2c –c informentry –inform –i 10.129.175.80 –add

A SNMPv1/v2c entries IP Address, Community Name, and Type can be changed. The following command is used to change a SNMPv2c entry’s Community Name, IP Address, and Type:

snmptrapconfig –v 2c –c newtrapentry –o informentry –i 10.129.178.33 –old_ipaddr 10.129.175.80 --trap –old_type inform –change

The following command is used to delete a SNMPv2c inform entry:

snmptrapconfig –v 2c –c newtrapentry –i 10.129.178.33 –trap –del

6.2.2 SNMPv3

SNMPv3 entries can be configured using the *snmptrapconfig* command with the following options:

-v	Version 3
-u	User Name - A string with a minimum length of 1 and a max length of 31. It can contain alphanumeric characters. However certain charters such as ‘`’ or ‘\’’or ‘\\’ or ‘&’ or ‘,’ or ‘ ‘ or ‘=’ or ‘;’ or ‘””’
--trap/--inform	Trap type – Trap (SNMPv1 and SNMPv2v) or inform (SNMPv2c)
-e	Engine ID (Informs only)
-i	A valid IP Address
--port	Port - Allows users to change the destination port (default is 162)
-s	Security Model – CM supports authNoPriv and authPriv
-a	Authentication Password – An alphanumeric string with a minimum length of 8 and a max length of 64. The password cannot contain certain character such as ‘`’ or ‘\’’or ‘\\’ or ‘&’ or ‘,’ or ‘ ‘or ‘”” or ‘-‘
--auth_prot	Authentication Protocol MD5 or SHA.
-p	Privacy Password – An alphanumeric string with a minimum length of 8 and a max length of 64. The password cannot contain certain characters such as ‘`’ or ‘\’’or ‘\\’ or ‘&’ or ‘,’ or ‘ ‘or ‘”” or ‘-‘
--priv_prot	Privacy Protocol of DES or AES128
-o	Old User Name (change command only)
–old_ipaddr	Old IP Address (change command only)
--old_type	Old Type (change command only)
--add	Add an entry.
--change	Change an entry.
--delete	Delete an entry

SNMPv3 entries to be added, changed, and deleted using the *snmptrapconfig* command as follows:

The following command is used to add a SNMPv3 trap entry:

snmptrapconfig –v 3 –i 10.129.178.30 –u trapname –trap -s authNoPriv –a authenitication pw –auth_prot MD5 –add

The following command is used to add a SNMPv3 inform entry:

snmptrapconfig –v 3 –i 10.129.178.30 –u informname –inform -s authNoPriv –a informauthenitication pw –auth_prot MD5 –e farendengineid –add

A SNMPv3 user name, IP Address, Type and passwords can be changed. The following command is used to change SNMPv3 entries IP Address:

snmptrapconfig –v 3 –i 10.129.178.66 –old_ipaddr 10.129.178.30 –u trapname –s authNoPriv –a authenticationpw –auth_prot MD5 –change

The following command is used to change a SNMPv3 authentication password:

snmptrapconfig –v 3 –i 10.129.178.66–u trapname –s authNoPriv –a newauthenticationpw –auth_prot MD5 –change

The following command is used to delete a SNMPv3 entry:

snmptrapconfig –v 3 –i 10.129.178.66–u trapname –s authNoPriv –a newauthenitication pw –auth_prot MD5 –del

6.3 snmpinctrapconfig command

CM not only sends traps but it can receive traps. INADS Alarming and h.248 Media Gateway's send traps to CM. In previous releases of CM incoming trap authorization checking was disabled. Therefore CM could not receive SNMPv3 traps. In addition, it would process any SNMPv1 and SNMPv2 traps that were sent to it. To make CM's SNMP trap receiver more secure, a new *snmpinctrapconfig* command was developed and allows users to configure SNMPv1, SNMPv2c, and SNMPv3 incoming trap authorization. By default incoming trap authorization is disabled. Before any incoming trap entries are added the *snmpinctrapconfig –enable* command must be executed. **Important Note** - Incoming trap authorization checking will impact INADS alarming; therefore you must add an incoming trap entry(s) to allow INADS community string(s). If you do not, INADs alarming will not work properly. Use the almsnmpconf command to determine incoming INADS community name string(s). The *snmpinctrapconfig –disable* command will disable incoming trap authorization checking and delete any configured incoming trap administration. Up to 250 incoming trap entries are supported.

As with the other commands, the createUser directive is used to create SNMPv3 users so that the authentication and privacy passwords are stored encrypted. The minimum security model supported by the snmptraprconfig command is authNoPriv. CM does not support a security model of noAuthNoPriv. Usage for the s*nmpinctrapconfig* command is as follows:

```
server1> snmpinctrapconfig --help
usage:
  snmpinctrapconfig -v 1|2 -c <comm_name> [-o <comm_name>]
                --add | --delete | --change
  snmpinctrapconfig -v 3  --inform|--trap
```



```
            -u <user> [-o <user>]
            [-s authNoPriv | authPriv]
            [-a <auth_passwd>] [--auth_prot <protocol>]
            [-p <priv_passwd>] [--priv_prot <protocol>]
            --add | --delete | --change
 snmpinctrapconfig --help | -h
  whereas:
               required when changing the destination for an entry).
   -v          : SNMP version (1, 2, 3)
   --inform    : inform entry (v3 only)
   --trap      : trap entry (v3 only)
   --old_type  : previous entry type (inform / trap) (v3 only)
              (only valid with the 'change' option,
               required when changing the type for an entry).
   -c          : v1/v2c community name
   -u          : v3 user name
   -o          : previous community/user name
              (only valid with the 'change' option,
               required when changing the user/community name).
   -s          : v3 security model (authNoPriv, authPriv)
   -a          : v3 authentication password
   --auth_prot : v3 authentication protocol (MD5, SHA)
   -p          : v3 privacy password
   --priv_prot : v3 privacy protocol (none, DES, AES128)
   -e          : Engine ID
   --add       : add entry
   --change    : change entry
   --delete    : delete entry
   --auth_list : list supported authentication protocols.
   --priv_list : list supported privacy protocols.
   --disable   : Disable Incoming Trap feature.
   --enable    : Enable Incoming Trap feature.
   --rclist    : list possible return codes from this command.
   --rccode    : obtain error string associated with a return code.
   --help|-h   : usage (this)

DEFAULT: 'snmpinctrapconfig' called with no arguments will display the current
        configuration.

NOTE: Using the -a and -p option with arguments on the command line presents a
      security risk as the passwords may be logged. Use the -a and -p options
      without an argument to prevent this.
```

6.3.1 SNMPv1/v2c

SNMPv1 and SNMPv2c use the same Net-SNMP directive (in the snmptrapd.conf file). This directive does not check SNMP versions or type. It just checks Community Names. Therefore, both SNMPv1 and SNMPv2 versions are accepted by the snmptrapd when either version is added. SNMPv1 and SNMPv2c incoming trap entries can be configured using the *snmpinctrapconfig* command with the following options:

--enable/--disable	Enable/Disable incoming trap authorization checking.
-v	Version 1 or 2c
-c	Community Name – A string with a minimum length of 1 and a max length of 31. It can contain alphanumeric characters. However certain charters such as ‘`’ or ‘\’’or ‘\\’ or ‘&’ or ‘,’ or ‘ ‘ or ‘=’ or ‘;’ or ‘”’
-o	Old Community Name (change command only)
--add	Add an entry.
--change	Change an entry.
--delete	Delete an entry

SNMPv1/v2c entries to be added, changed, and deleted using the *snmpinctrapconfig* command as follows:

The following command is used to add a SNMPv1/v2c incoming trap/inform entry:

snmpinctrapconfig –v 1 –c public --add

A SNMPv1/v2c entry’s IP Address, Community Name, and Type can be changed. The following command is used to change a SNMPv1/v2c entry:

snmpinctrapconfig –v 2c –c private –o public --change

The following command is used to delete a SNMPv2c inform entry:

snmpinctrapconfig –v 2c –c private –o public --del

6.3.2 SNMPv3

Each SNMPv3 incoming trap/inform entry must have a unique user name. SNMPv3 entries can be configured using the *snmptinctrapconfig* command with the following options

--enable/--disable	Enable/Disable incoming trap authorization checking.
-v	Version 3
-u	User Name - A string with a minimum length of 1 and a max length of 31. It can contain alphanumeric characters. However certain charters such as ‘`’ or ‘\’’or ‘\\’ or ‘&’ or ‘,’ or ‘ ‘ or ‘=’ or ‘;’ or ‘”’
--trap/--inform	Trap type - Trap (SNMPv1 and SNMPv2v) or inform (SNMPv2c)
-e	Engine ID – Engine ID of sending system (Traps only)
-s	Security Model – CM supports authNoPriv and authPriv
-a	Authentication Password – An alphanumeric string with a

	minimum length of 8 and a max length of 64. The password cannot contain certain characters such as ‘`’ or ‘\’’or ‘\\’ or ‘&’ or ‘,’ or ‘ ‘or ‘”’ or ‘-‘
--auth_prot	Authentication Protocol MD5 or SHA.
-p	Privacy Password - An alphanumeric string with a minimum length of 8 and a max length of 64. The password cannot contain certain characters such as ‘`’ or ‘\’’or ‘\\’ or ‘&’ or ‘,’ or ‘ ‘or ‘”’ or ‘-‘
--priv_prot	Privacy Protocol of DES or AES128
-o	Old User Name (change command only)
--old_type	Old Type (change command only)
--add	Add an entry.
--change	Change an entry.
--delete	Delete and entry

SNMPv3 entries can be added, changed, and deleted using the *snmpinctrapconfig* command as follows:

The following command is used to add a SNMPv3 incoming trap entry:

snmpinctrapconfig –v 3 —u snmpinctrapname --trap -s authNoPriv –a authenitication pw –auth_prot MD5 -e farendengineid --add

The following command is used to add a SNMPv3 incoming trap entry:

snmpinctrapconfig –v 3 —u snmpincinformname --inform -s authNoPriv –a authenitication pw –auth_prot SHA --add

A SNMPv3 user name, type and passwords can be changed. The following command is used to change SNMPv3 entries user name:

snmpinctrapconfig –v 3 –u newinctrapname –o snmpinctrapname -s authNoPriv –a authenitication pw –auth_prot MD5 --change

The following command is used to change a SNMPv3 authentication password:

snmpinctrapconfig –v 3 –u newinctrapname -s authNoPriv –a newauthenitication pw –auth_prot MD5 –change

The following command is used to delete a SNMPv3 incoming trap entry:

snmpinctrapconfig –v 3 –u newinctrapname -s authNoPriv –a newauthenitication pw –auth_prot MD5 –del

6.4 snmpconfig command

The *snmpconfig* command displays the systems local Engine ID. Furthermore it allows users to reset the system's snmpd.conf and/or snmptrapd.conf files back to their original default state. When used, this command will erase all data from the current SNMP configuration files. Usage for the *snmpconfig* command is as follows:

```
server1> snmpconfig
Error: Invalid usage... (command line parsing)
usage:
  snmpconfig -e -- Retrieve local Engine ID
  snmpconfig -r [both, snmpd, snmptrapd] -- Reset Configuration files to default
  snmpconfig -h -- display USAGE
```

6.5 snmpagentsstatus command

The *snmpagentsstatus* command displays the status of the CM SNMP processes. When no options are added the command will return the status of all the CM SNMP processes. When the -s or --s or --status option is added along with an SNMP process name, only the that process is displayed. The usage for the *snmpagentsstatus* command is as follows:

```
server1> snmpagentsstatus --help
usage:
  snmpagentsstatus --s [process]
  snmpagentsstatus --help | -h
   where:
    --status [process]   : check status of all or specified process
    --rclist             : list possible return codes from this command.
    --help | -h          : usage (this)
```

The following is an example of the *snmpagentsstatus* command without any options:

```
Server1> snmpagentsstatus
MasterAgent(snmpd):      UP
CMSubAgent(cmsubagt):    UP
CMLoadAgent(cmldagt):    UP
CMFPAgent(cmfpagt):      UP
SNMPManager(snmptrapd):  UP
```

The following is an example of a single process query:

```
Server1> snmpagentsstatus --status CMSubAgent
CMSubAgent(cmsubagt):    UP
```

6.6 testcustalm command

The *testcustalm* command sends a FP Test Trap to configured trap destinations.
The usage for the *testcustalm* command is as follows:

```
server1> testcustalm --help
```


Usage: testinads | testcustalm
No command line parameter required. It tests the health of alarm processing between the system and the receiver.

The following is an example of the testcustalm command:

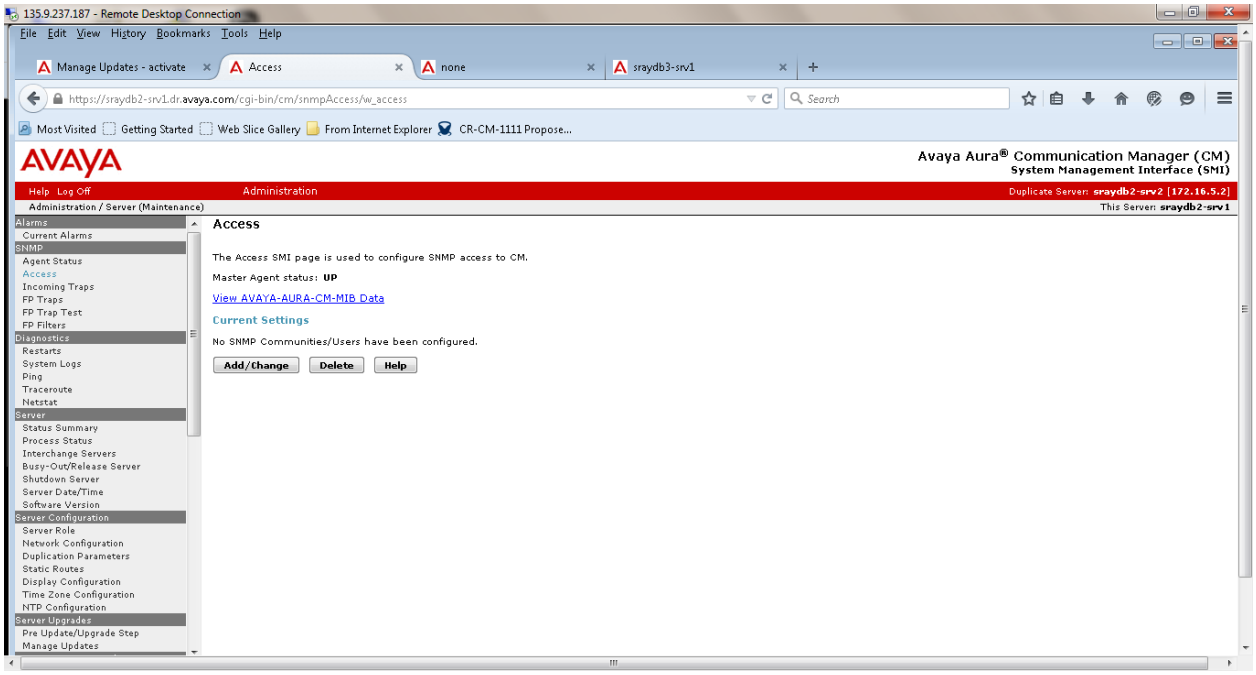
```
Server1> testcustalm
Note: The Application is waiting for a response from CommunicaMgr.
                *** Be Patient ***

Reply from CommunicaMgr: Test message was successfully reported to the GMM.
```

7.0 SNMP SMI Pages

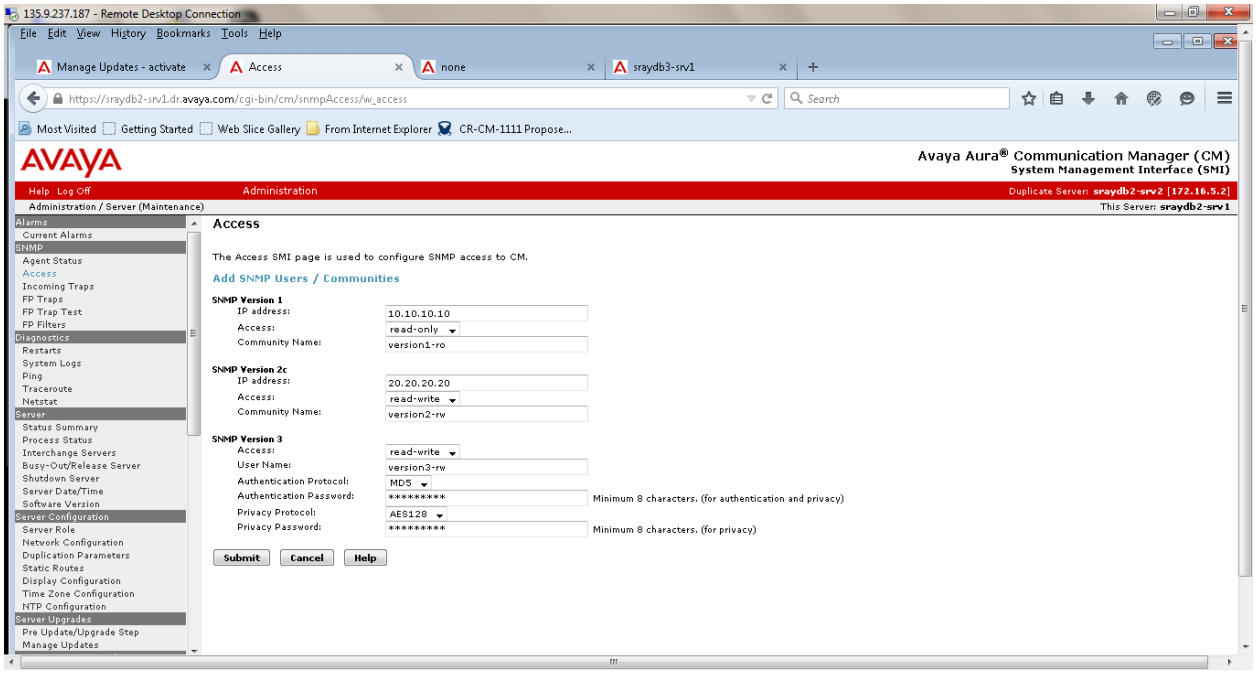
In addition to using SNMP get and set commands, users can also configure SNMP using the SNMP SMI Pages. The SNMP SMI Pages allow users to add, change and delete SNMP Access, FP Traps, Incoming Traps, and FP Filters configuration data.

7.1 SNMP Access SMI Page

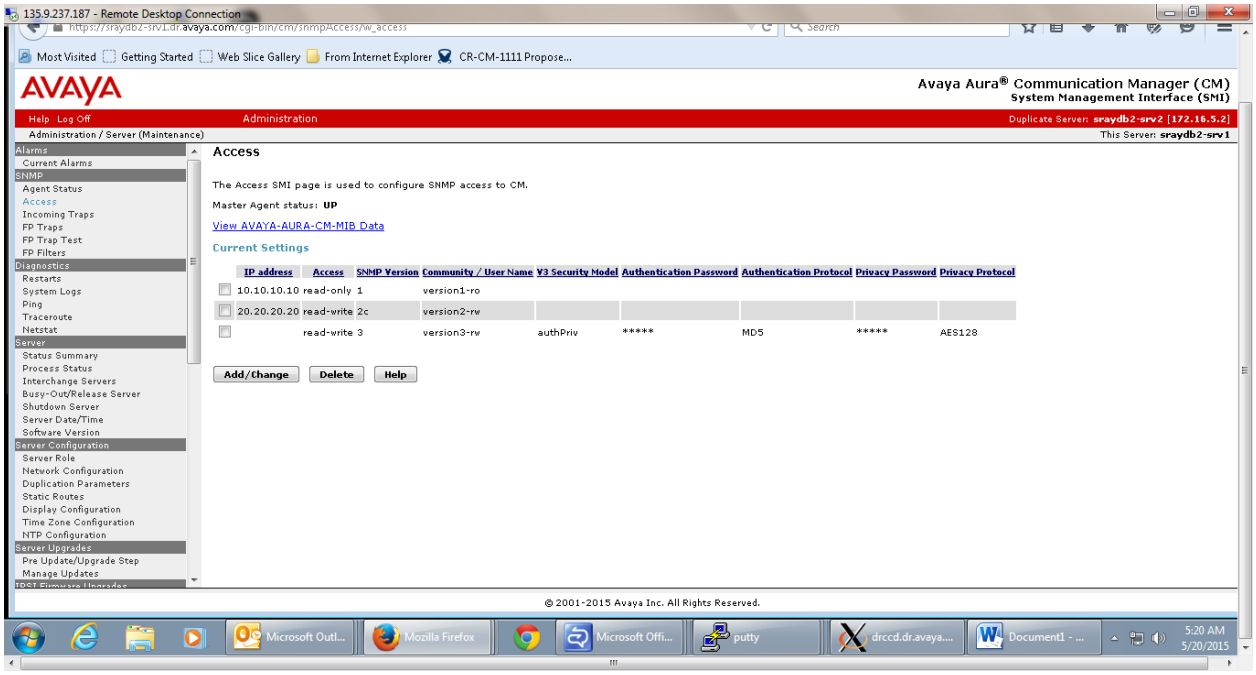
The *CM SNMP Access SMI Page* administers the same access configuration data as the *snmpuserconfig* command. Configuration data entered by the *snmpuserconfig* command is displayed on the *CM SNMP Access SMI Page* and configuration data entered on the *CM SNMP Access SMI Page* is displayed by the *snmpuserconfig* command. As with the *snmpuserconfig* command the SNMP Access page allows users to add, change, and delete SNMP access data. In addition, the CM SNMP Access SMI Page provides a link to the AVAYA-AURA-CM-MIB.txt. Configuring SNMP access via *the CM SNMP Access SMI Page* is simple. To add a user select the add/change button.



Three SNMP entries (One of each version) can be added at the same time. The following example shows a SNMPv1, a SNMPv2, and a SNMPv3 being added at the same time:

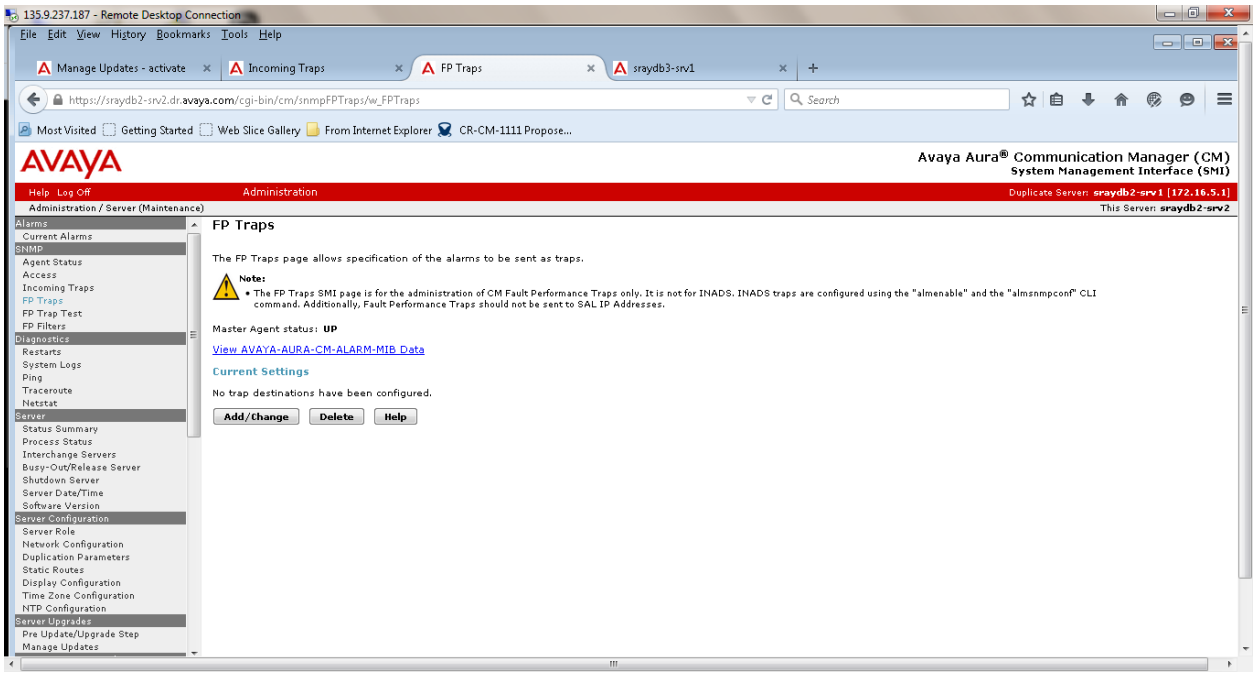


After inputting the access information, select submit. The following example displays the access data that was just submitted.

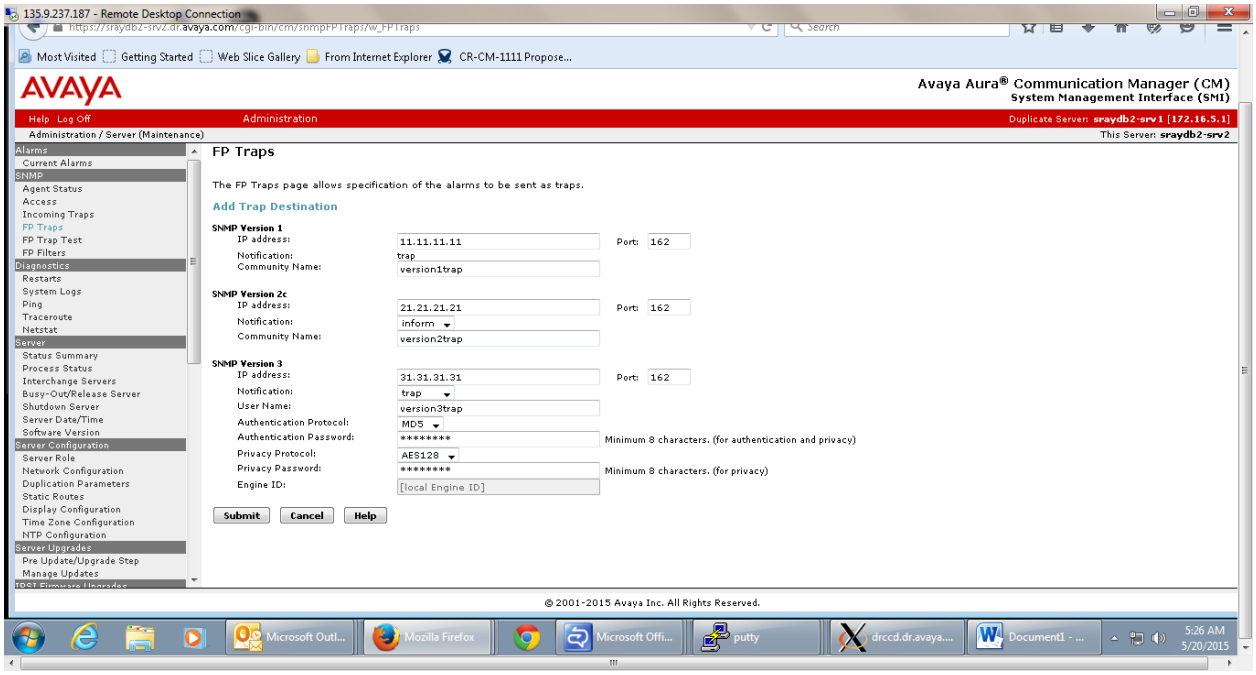

135.9.237.187 - Remote Desktop Connection
Most Visited
Getting Started
Web Slice Gallery
From Internet Explorer
CR-CM-1111 Propose...
AVAYA
Avaya Aura® Communication Manager (CM)
System Management Interface (SMI)
Help Log Off
Administration
Duplicate Server: sraydb2-srv2 [172.16.5.2]
Administration / Server (Maintenance)
This Server: sraydb2-srv1
Alarms
Current Alarms
SNMP
Agent Status
Access
Incoming Traps
FP Traps
FP Trap Test
FP Filters
Diagnostics
Restarts
System Logs
Ping
Traceroute
Netstat
Server
Status Summary
Process Status
Interchange Servers
Busy-Out/Release Server
Shutdown Server
Server Date/Time
Software Version
Server Configuration
Server Role
Network Configuration
Duplication Parameters
Static Routes
Display Configuration
Time Zone Configuration
NTP Configuration
Server Upgrades
Pre Update/Upgrade Step
Manage Updates
Access
The Access SMI page is used to configure SNMP access to CM.
Master Agent status: UP
View AVAYA-AURA-CM-MIB Data
Current Settings
IP address
Access
SNMP Version
Community / User Name
V3 Security Model
Authentication Password
Authentication Protocol
Privacy Password
Privacy Protocol
10.10.10.10 read-only 1 version1-ro
20.20.20.20 read-write 2c version2-rw
read-write 3 version3-rw authPriv ***** MD5 ***** AES128
Add/Change
Delete
Help
© 2001-2015 Avaya Inc. All Rights Reserved.
Microsoft Outl...
Mozilla Firefox
Microsoft Offi...
putty
drccd.dr.avaya....
Document1 - ...
5:20 AM
5/20/2015

7.2 SNMP FP Traps SMI Page

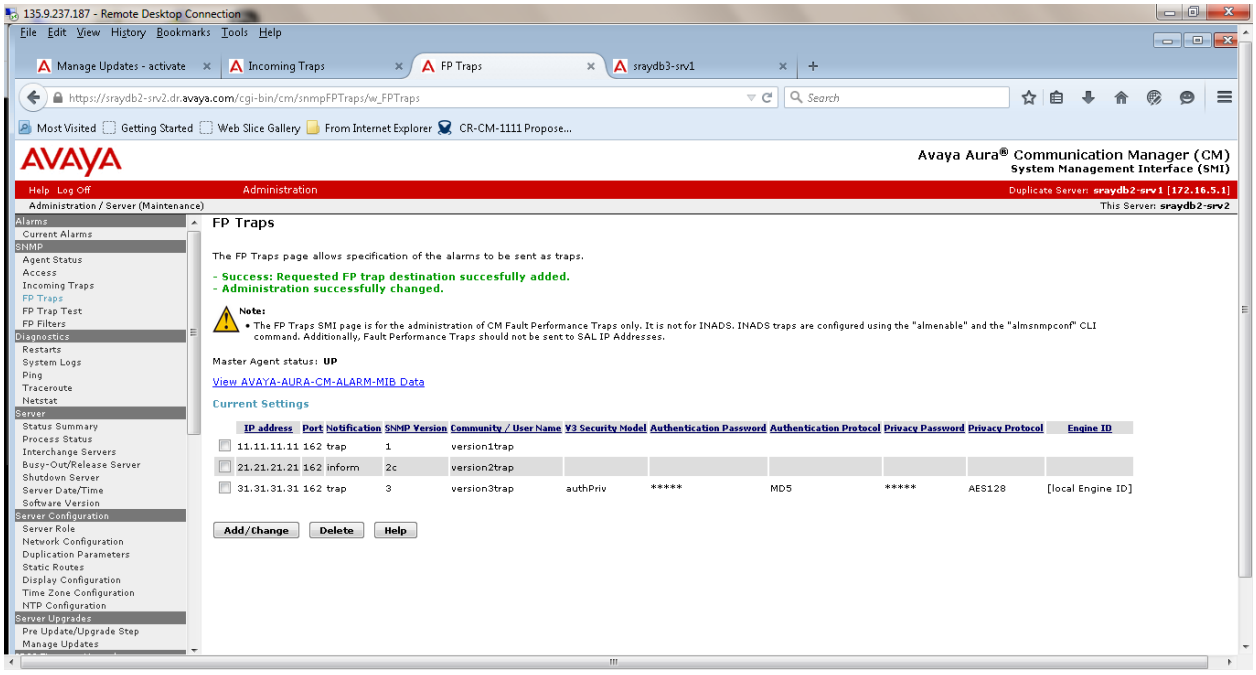
The *CM SNMP FP Traps SMI Page* administers the same configuration data as the *snmptrapconfig* command. Configuration data entered by the *snmptrapconfig* command is displayed on the *CM SNMP FP Traps SMI Page* and configuration data entered on the *CM SNMP FP Traps SMI Page* is displayed by the *snmptrapconfig* command. As with the *snmptrapconfig* command the SNMP Incoming Trap page allows users to add, change, and delete SNMP incoming traps data. To add a user select the add/change button.



Three SNMP entries (One of each version) can be added at the same time. The following example shows a SNMPv1, a SNMPv2, and a SNMPv3 being added at the same time:



After inputting trap information, select submit. The following example displays the trap data that was just submitted.

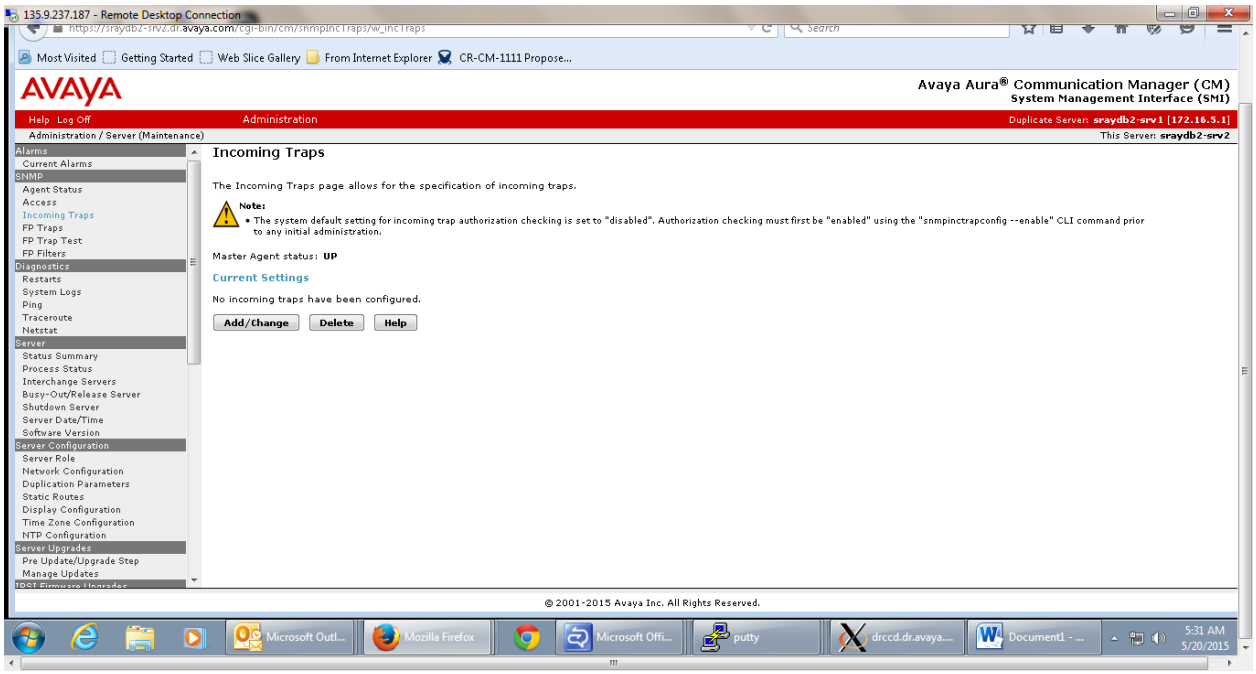


7.3 SNMP Incoming Traps SMI Page

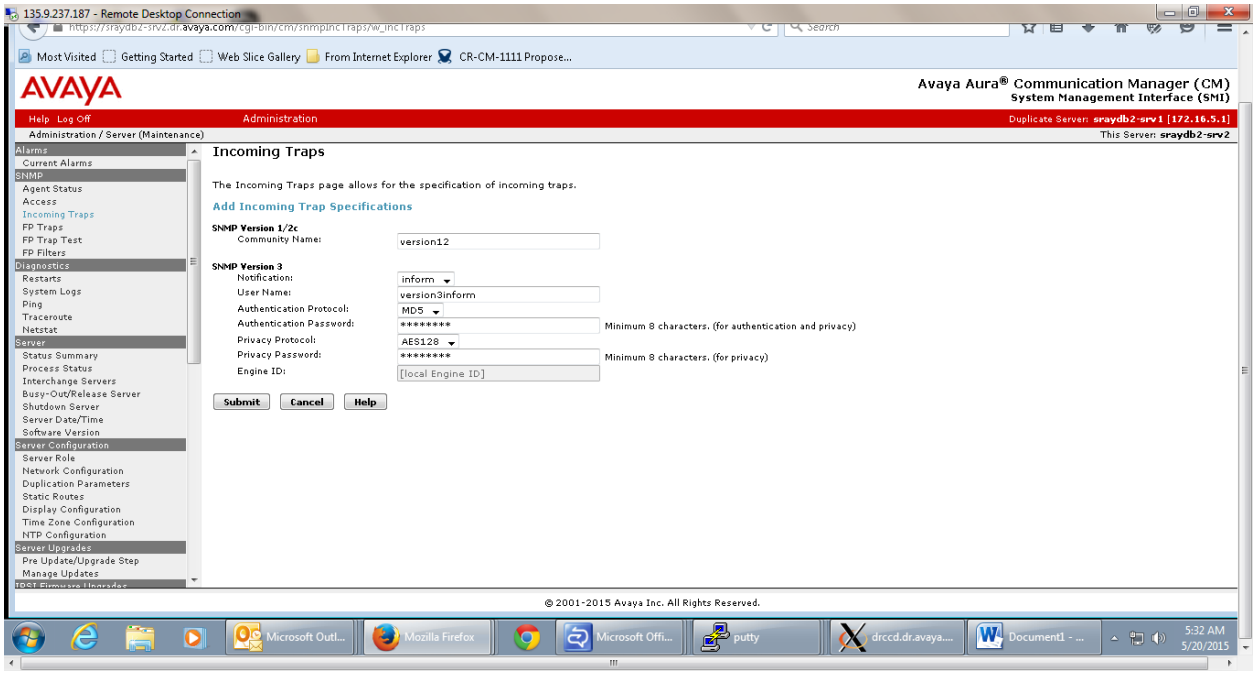
The *CM SNMP Incoming Traps SMI Page* administers the same configuration data as the *snmpinctrapconfig* command. Configuration data entered by the *snmpinctrapconfig*

command is displayed on the *CM SNMP Incoming Traps SMI Page* and configuration *snmpinctrapconfig* command. As with the *snmpinctrapconfig* command the SNMP Incoming Trap page allows users to add, change, and delete SNMP incoming traps data. To add a user select the add/change button.

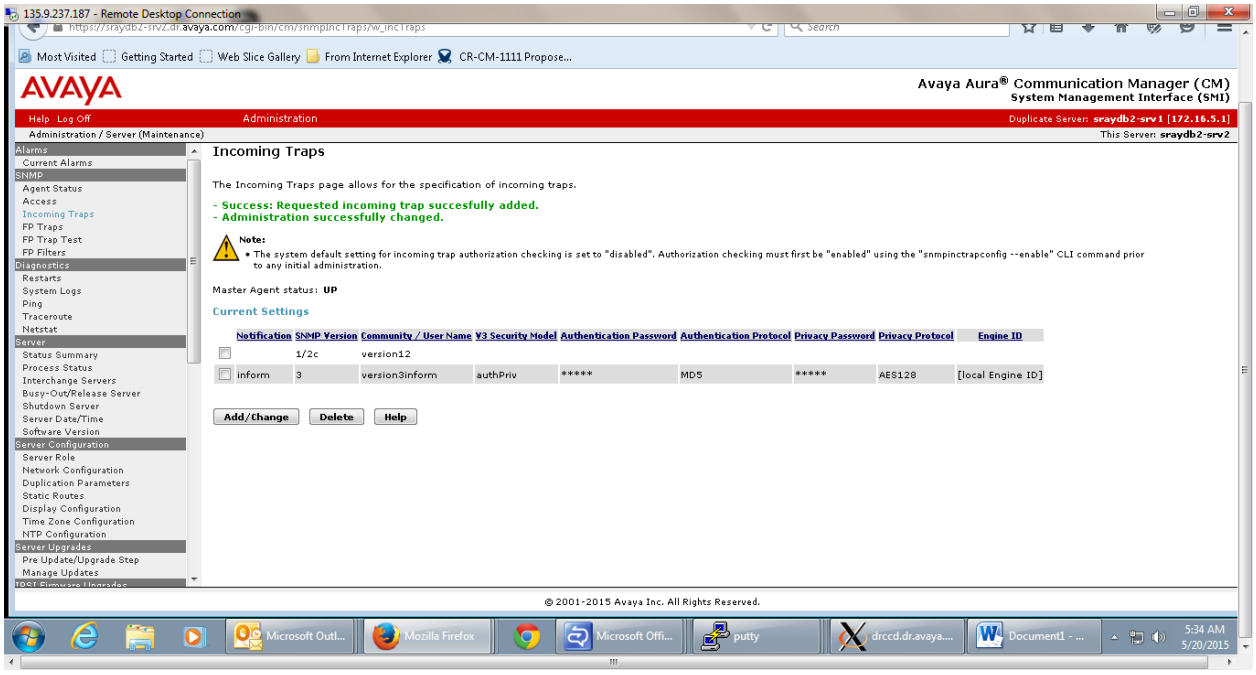
NOTE: The Incoming Trap Authorization is disabled by default and must be enabled via the *snmpinctrapconfig –enable* command before any incoming trap entries be configured.



Two SNMP entries (One for SNMPv1/SNMPv2 and one for SNMPv3) can be added at the same time. The following example shows a SNMPv1/ SNMPv2, and a SNMPv3 being added at the same time:



After inputting trap information, select **Submit**. The following example displays the incoming trap data that was just submitted.



7.4 SNMP FP Filters SMI Page

The *CM SNMP FP Filters SMI Page* administers alarm filters for FP Traps. Filters can also be administered using SNMP get and SNMP set commands (see section 4.2.2.1). Configuration data entered by using SNMP get and SNMP set commands is displayed on the *CM SNMP Incoming Traps SMI Page* and configuration data entered on the *CM SNMP Incoming Traps SMI Page* can be displayed by using SNMP get commands. The *SNMP FP Filters SMI* page allows users to add, change, and delete SNMP FP Filters. A default Active Major Minor Communication Manager Alarm Trap is added at initial installation. Only Communication Manager Alarm Traps are supported by the SNMP FP Filters SMI Page. Platform/Server and Restart Notification Alarm Trap filters can only be configured using SNMP get and SNMP set commands (see sections 4.2.2.2 and 4.2.2.3). To delete or change a single filter entry select the + button next to the entry. To add a new filter, select the **Add** button. To delete all Communication Manager Filters select the **Delete All** button.

NOTE: This page is slow rendering and may take up to 30 seconds to display current filter information. Also, to see any changes that have been made, the page must be reloaded.