ANACOM, INC.

1996 Lundy Ave, San Jose, CA 95131, USA
Phone: 408.519.2062 Fax: 408.519.2063
www.anacominc.com

SNMP COMMAND

SNMP (Rev 78)

This command serves to list all SNMP configuration parameters, but it can also be used to set them. With no argument, basic usage is given.

Example:

```
SNMP
SNMP [HELP] [COMMUNITY SYSCONTACT SYSLOCATION SYSNAME SYSOBJECID/OID TRAPS LIST]
```

SNMP LIST (Rev 78)

All SNMP configuration parameters will be displayed with the LIST argument, (valid aliases are ALL and SHOW.)

Example:

```
SNMP LIST
SNMP COMMUNITY public
SNMP sysOBJECTID [10] 1 3 6 1 4 1 4578 6 1 1
SNMP sysCONTACT Anacom support <techsupport@anacominc.com>
SNMP sysNAME www.anacominc.com
SNMP sysLOCATION I am in my office!
SNMP TRAPS 30   192.168.1.11,23   0.0.0.0,23   0.0.0.0,23   0.0.0.0,23
```

Valid usage:

```
SNMP ALL/SHOW/LIST                      see above
SNMP HELP  [help topic]                 see below
SNMP COMMUNITY  [community string]      see below
SNMP SYSCONTACT  [sysContact string]    see the SYSCONTACT command
SNMP SYSLOCATION  [sysLocation string]  see the SYSLOCATION command
SNMP SYSNAME  [sysName string]          see the SYSNAME command
SNMP SYSOBJECTID  [sysObjectID string]  see the OID command
SNMP TRAPS  [traps configuration]       see the SNMPT command
SNMP TRAPS CLEAR  [ALL | trap index]    see the SNMPT command
```

SNMP HELP can return help information for any of the valid arguments that can be give, such as SYSCONTACT.

Example:

```
SNMP HELP sysCONTACT
SNMP sysCONTACT [string] - get/set the system OID contact information
```

SNMP COMMUNITY [Community_string] (Rev 78)

Previous to revision 78, the firmware's SNMP V1 agent did not check the community string. Beginning with firmware revision 78, it does. Public is always accepted, but is now restricted to read-only access to data. When this command is used to set an alternate to public, then the new community string can be used for read/write access.

Example:

```
SNMP COMMUNITY demoprivate
```

SNMP OID / SYSOBJECTID (Rev 78)

The unit's system Object ID (OID) that is used in SNMP to identify the MIB data that can be accessed can be displayed for reference, as shown below.

Example:

```
SNMP OID
SNMP OID [10] 1 3 6 1 4 1 4578 6 1 1


SNMP OID RAW
SNMP OID RAW 2B 6 1 4 1 A3 62 6 1 1
```

SNMP SYSCONTACT [contact_info] (Rev 78)

A standard system OID, system.sysContact=1.3.6.1.2.1.1.4.0, a data string that gives contact information should the user have a problem. In the case of AnaCom units, the default is the Anacom, Inc. email support address techsupport@anacominc.com, this can be changed in the field however.

Example:

```
SNMP SYSCONTACT
SNMP SYSCONTACT Anacom support <techsupport@anacominc.com>
```

SNMP SYSDESCR (Rev 78)

A standard system OID, system.sysDescr=1.3.6.1.2.1.1.1.0, a data string that gives a basic description for the unit. In the case of AnaCom units, we are returning the comma-delimited data that would be returned from the **INFO** command: Product label, Serial#, HW part#, HW rev#, SW rev#, user defined label

Example:

```
sysdescr
40W C-Band BUC,SN 999995,PN 12345,HWREV 002,SWREV 00078,LABEL booboo
```

Note: this command is read-only.

SNMP SYSLOCATION [location] (Rev 78)

A standard system OID, system.sysLocation=1.3.6.1.2.1.1.6.0, a data string for keeping the unit's location. The default is “Undisclosed.”

Example:

```
syslocation I am here!
```

SNMP SYSNAME [name] (Rev 78)

A standard system OID, system.sysName=1.3.6.1.2.1.1.5.0, a data string for keeping the unit's SNMP name. The default is **www.anacominc.com**.

SNMP TRAPS [ON | OFF] (Rev 78)

SNMP V1 traps are supported beginning with firmware revision 72 using periodic trap reporting. A simple message can be returned to up to four different clients (using their IP addresses) on a periodic basis. The actual trap message used for each client can be different. The default trap message is ALARMS, represented by MIB variable 23 for BUC and XCVR, 14 for a PS. The periodic time for trap transmission however is the same for all clients. The default periodic time is 30 seconds.

Event-driven trap reporting was introduced in firmware revision 78, and returns the ALARMS data to the specified targets. This is now the default trap reporting behavior as well - the ALARMS data is sent as an event-driven trap when the alarm state of the unit changes.

This command can turn traps ON and OFF.

Example:

```
SNMP TRAPS
SNMP TRAPS OFF  0.0.0.0  192.168.1.3  0.0.0.0  0.0.0.0


SNMP TRAPS ON


SNMP TRAPS
SNMP TRAPS ON_EVENT  0.0.0.0  192.168.1.3  0.0.0.0  0.0.0.0
```

SNMP TRAPS ADDR [IP address … IP address] (Rev 78)

Set up to four IP addresses to be the targets of traps for the SNMP agent running in this unit. To see what targets are set, use the command **SNMP LIST**.

Note: It is not necessary to type all the IP address from scratch when this command is used. A dash "-" symbol can be used to leave a previously set target unchanged.

Example:

```
SNMP TRAPS
SNMP TRAPS OFF  192.168.1.3  0.0.0.0  0.0.0.0  0.0.0.0


SNMP TRAPS ADDR - 192.168.1.34


SNMP TRAPS
SNMP TRAPS OFF  192.168.1.3  192.168.1.34  0.0.0.0  0.0.0.0
```

Note: If the ODU/PS is using a static IP address, it may be important to manually set the default gateway address correctly in order to route the trap messages to the targets. See the ODU command **IP** in the ODU Ethernet Configuration User Guide.

Example:

```
IP  ADDR  192.168.1.27
```

```
IP  DEFAULT  192.168.1.1
IP  NETMASK  255.255.255.0
SAVE
```

Note: if the target is not running an SNMP trap listener, it might return an ICMP message back to the Anacom, Inc. device indicating that the SNMP port is unreachable. This won't cause any problems, but the ICMP message will not be acted upon by the firmware.

Note: In order to successfully send a trap to a local target, one that is in the same IP subnet as the device sending traps, it is necessary to have first captured the Ethernet MAC address of that device. The MAC address of a new listener will not be requested using an ARP Request Ethernet packet until it is time to send a trap. The list of MACs for known devices can be shown using the **ARP** command, no arguments.

Example:

```
ARP
IP Address        MAC (H/W) Address
===============   =================
192.168.1.1       54:75:D0:26:9F:E5
192.168.1.59      80:FA:5B:02:CB:7A
192.168.1.161     00:90:27:89:F6:AF
192.168.1.226     00:14:22:F1:2F:BF
192.168.1.229     00:25:11:51:E7:6A
===============   =================
```

If there should be a problem, and the MAC address associated with a trap listener's IP address is incorrect, the ARP table can be cleared.

Example:

```
ARP CLEAR
```

Note: Sending traps to local targets on a LAN that are assigned IP addresses using a DHCP server could be problematic, as those devices might be assigned a different IP address at some point, if they are not continuously active all the time.

Note: beginning with firmware revision 89, it is possible delete a specific ARP entry based in IP address.

Example:

```
ARP DEL 192.168.1.59
```

Note: The ARP table can hold a maximum of eight entries, so if M&C activity of the unit requires more than that, there might be a noticeable slowdown in Ethernet response.

SNMP TRAPS CLEAR [Index] (Rev 78)

The trap configuration associated with a particular index, (counting from zero,) can be erased, leaving the rest unaffected. Alternatively, all the targets can be erased.

Example:

```
SNMP TRAPS
SNMP TRAPS 30  192.168.1.11,23  192.168.1.68,23  0.0.0.0,23  0.0.0.0,23

SNMP TRAPS CLEAR 0
```

```
SNMP TRAPS
SNMP TRAPS 30  0.0.0.0,23  192.168.1.68,23  0.0.0.0,23  0.0.0.0,23

SNMP TRAPS CLEAR

SNMP TRAPS
SNMP TRAPS OFF  0.0.0.0  0.0.0.0  0.0.0.0  0.0.0.0
```

SNMP TRAPS SEND (Rev 89)

Send a trap message to all targets that have been defined using the **SNMP TRAPS ADDR** command. This will be useful in testing that targets are properly configured.

Note: this feature was not implemented until firmware revision 89.

SNMP TRAPS MIB_VAR [var … var] (Rev 78)

Beginning with firmware revision 78, the default agent behavior is to send the ALARMS data as an event-driven trap. It is still possible however as an alternative to this, to periodically send arbitrary data specified in the MIB.

This is now done using this **SNMP** command syntax, where each variable specified matches the IP address target configured using **SNMP TRAPS ADDR**.

This configuration will not take effect however until the default operation is changed to periodic trap reporting using the **PSEC** syntax.

Note: as with the **SNMP TRAPS ADDR** syntax described above, a dash "-" symbol can be used as an empty placeholder for a target that we do not want to retype or change.

Example:

```
SNMP TRAPS PSEC 30


SNMP TRAPS ADDR - 192.168.1.34


SNMP TRAPS
SNMP TRAPS 30  0.0.0.0,23  192.168.1.34,23  0.0.0.0,23  0.0.0.0,23


SNMP TRAPS MIB_VAR - 9


SNMP TRAPS
SNMP TRAPS 30  0.0.0.0,23  192.168.1.34,9  0.0.0.0,23  0.0.0.0,23
```

In the above example (somewhat contrived,) we will now send the up-time since last reset for this unit to target 192.168.1.34 every 30 seconds.

Note: Beginning with firmware revision 78, if this feature is used and traps are turned off, **SNMP TRAPS OFF**, and then turned back on, **SNMP TRAPS ON**, trap generation will default back to event-driven behavior.

Note: this is a feature we do not expect to get used much.

SNMP TRAPS PSEC [Time in seconds] (Rev 78)

Sets SNMP trap generation to periodic reporting and sets the periodic trap transmission time in seconds. The minimum time between transmissions is 30 seconds. There is no maximum limit.

Example:

```
SNMPT PSEC 30
```

Note: this command sets the value, it does not return data if no argument is given.

Testing SNMP Traps

The sending and successful reception of SNMP traps by trap listeners can be tested by causing a temporary alarm in the ODU or protection switch. A simulated alarm can be generated using the **SIMFAIL** command,

Example:

```
SIMFAIL ON
```

This will generate a SIMFAIL alarm that will last approximately 30 seconds. It's activation can be seen in Supervisor or using the **ALARMS** command. If event-driven traps, or periodic traps with the ALARM MIB variable is set, then the targets specified to receive SNMP traps will get a trap message with the SIMAIL alarm set. After 30 seconds this alarm will canceled and another trap should be sent showing that it has been cleared.

Beginning with firmware revision 89 for ODU/PS devices, it is possible to send a trap message at will.

Example:

```
SNMP TRAPS SEND
```

This will send a trap immediately, using whatever configuration has been previous set using the SNMP TRAPS command.

Note: a specified target could fail to catch a trap message because it is not running a trap listener process; if the target is local, the sending device could not get the MAC address of the target, or if the target is not local, the default gateway was not properly specified, or the router managing the LAN was not able to route the trap messages to the specified targets, etc.

Monitoring using Net-SNMP

There are commercial NMS tools for monitoring AnaCom, Inc. outdoor equipment using SNMP. This section gives examples using open source console tools provided by Net-SNMP, see: http://www.net-snmp.org/. These examples were created using net-snmp tools via the command-line on a Linux workstation.

Example: walking the system OID for an AnaCom, Inc. device that turns out to be an ELSAT BUC:

```
$ snmpwalk -v 1 -c public 192.168.1.13
SNMPv2-MIB::sysDescr.0 = STRING: ELSAT 40EC BUC,SN 072555,PN 32028,LABEL
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.4578.6.1.1
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (8014200) 22:15:42.00
SNMPv2-MIB::sysContact.0 = STRING: Anacom support <techsupport@anacominc.com>
SNMPv2-MIB::sysName.0 = STRING: www.anacominc.com
SNMPv2-MIB::sysLocation.0 = STRING: Undisclosed
```

```
SNMPv2-MIB::sysServices.0 = INTEGER: 1
End of MIB
```

Example: walking the AnaCom MIB for an ELSAT BUC on a local network with the IP address 192.168.1.13, and an enterprise OID = enterprises.4578; a BUC uses the device sub ID .6.1.1:

```
$ snmpwalk -v 1 -c public 192.168.1.13 1.3.6.1.4.1.4578.6.1.1
SNMPv2-SMI::enterprises.4578.6.1.1.1 = STRING: "TX ELSAT "
SNMPv2-SMI::enterprises.4578.6.1.1.2 = STRING: "32028"
SNMPv2-SMI::enterprises.4578.6.1.1.3 = STRING: "072555"
SNMPv2-SMI::enterprises.4578.6.1.1.4 = STRING: "02"
SNMPv2-SMI::enterprises.4578.6.1.1.5 = STRING: "ELSAT 40EC BUC"
continues on with more data returned during the walk...
```

Example: to return the serial number for this device:

```
$ snmpget -v 1 -c public 192.168.1.13 enterprises.4578.6.1.1.3
SNMPv2-SMI::enterprises.4578.6.1.1.3 = STRING: "072555"
```

If the AnaCom, Inc. MIB has been placed in the proper folder, then data can be retrieved without resorting to using the OID numeric identifiers:

```
$ snmptranslate -On ANACOM-MIB::anaBUC
.1.3.6.1.4.1.4578.6

$ snmpget -v 1 -c public 192.168.1.13 ANACOM-MIB::anaBUCSerialNumber
ANACOM-MIB::anaBUCSerialNumber = STRING: "072555"
```

It is often more convenient to return data or walk the MIB, using snmpgetnext:

```
$ snmpgetnext -v 1 -c public 192.168.1.13 ANACOM-MIB::anaBUC
ANACOM-MIB::anaBUCOduType = STRING: "TX ELSAT "
```

All MIB-specified data can be returned for this device using snmpwalk:

```
$ snmpwalk -v 1 -c public 192.168.1.13 ANACOM-MIB::anaBUC
```

Example of walking the MIB for a 1+1 Protection Switch:

```
$ snmpwalk -v 1 -c public 192.168.1.30 ANACOM-MIB::anaPS1f1
ANACOM-MIB::anaPS1f1OduType = STRING: "PS1:1 "
ANACOM-MIB::anaPS1f1OduPartNumber = STRING: "32313"
ANACOM-MIB::anaPS1f1SerialNumber = STRING: "072925"
ANACOM-MIB::anaPS1f1OduRev = STRING: "82"
ANACOM-MIB::anaPS1f1Model = STRING: "ARM9 Protection Switch"
ANACOM-MIB::anaPS1f1Label = STRING: "none"
ANACOM-MIB::anaPS1f1MnCPartNumber = STRING: "33559"
ANACOM-MIB::anaPS1f1MnCRev = STRING: "00082"
ANACOM-MIB::anaPS1f1OnTime = STRING: "66285 (18 hours 24 min)"
ANACOM-MIB::anaPS1f1Age = STRING: "9168401 (106 days 2 hours 46 min)"
ANACOM-MIB::anaPS1f1Switch = STRING: "TXA:Auto RXA:Auto"
ANACOM-MIB::anaPS1f1StandbyMode = STRING: "HOT"
ANACOM-MIB::anaPS1f1AlarmLevel = STRING: "this feature is not available on this
device"
ANACOM-MIB::anaPS1f1Alarms = STRING: "CLEAR"
ANACOM-MIB::anaPS1f1Uname = STRING: "BUILT: Jun  8 2014, 17:25:04, STR912 Chip Rev
H, FLASH=SST"
ANACOM-MIB::anaPS1f1SNMPT = STRING: "TRAPS ON_EVENT  0.0.0.0  0.0.0.0  0.0.0.0
0.0.0.0"
```

```
ANACOM-MIB::anaPS1f1Temp = STRING: "34.6"
End of MIB
```

Note: It is of course necessary to already know the IP addresses of the units that we wish to monitor, this requires some means of walking all valid IP addresses within the space of a given mask.

Example script to return all Ethernet-equipped AnaCom, Inc devices on a local network:

```
sudo netdiscover -i eth0 -r 192.168.1.0/24 -P | grep 0c:ef:7c
```

Note: 0C:ef:7c is the designated Ethernet prefix for AnaCom devices. The suffix is comprised of each devices serial number.

Net-snmp can also be used to catch traps from remote equipment, the following example shows the capture of trap messages from an ODU to standard output, as it logs an alarm, and then when it is subsequently cleared:

```
root@ron-Linux64bit:/var/log# snmptrapd -f -Lo
NET-SNMP version 5.4.2.1
2014-06-10 15:29:21 192.168.1.13(via UDP: [192.168.1.13]:63415->[192.168.1.5])
TRAP, SNMP v1, community public
      SNMPv2-SMI::enterprises.4578.990 Enterprise Specific Trap (23) Uptime:
0:27:27.88
      SNMPv2-SMI::enterprises.4578.6.1.1.23 = STRING: "SIMFAIL"
2014-06-10 15:29:52 192.168.1.13(via UDP: [192.168.1.13]:63415->[192.168.1.5])
TRAP, SNMP v1, community public
      SNMPv2-SMI::enterprises.4578.990 Enterprise Specific Trap (23) Uptime:
0:27:28.19
      SNMPv2-SMI::enterprises.4578.6.1.1.23 = STRING: "CLEAR"
```

In the above example, the ODU has IP address 192.168.1.13, and the local machine running snmptrapd has IP address 192.168.1.5. In this example, **SIMFAIL** is a temporary simulated alarm that can be activated on an ODU using the command **SIMFAIL ON**.

Note: traps are usually caught by a system daemon listening on port 162. This example will not work if there is another daemon that was previously launched to catch traps, until it is killed.

In the following example, we start snmptrapd with all available MIBs that are loaded on they monitoring system, which in this case includes ANACOM-MIB:

```
$ sudo snmptrapd -f -Lo -m ALL
NET-SNMP version 5.4.2.1
2014-06-10 16:46:07 192.168.1.44(via UDP: [192.168.1.44]:63415->[192.168.1.5])
TRAP, SNMP v1, community public
      ANACOM-MIB::anaBUCTraps Cold Start Trap (23) Uptime: 0:00:12.63
      ANACOM-MIB::anaBUCAlarms = STRING: "SIMFAIL"
2014-06-10 16:46:38 192.168.1.44(via UDP: [192.168.1.44]:63415->[192.168.1.5])
TRAP, SNMP v1, community public
      ANACOM-MIB::anaBUCTraps Enterprise Specific Trap (ANACOM-
MIB::anaBUCAlarmTrap) Uptime: 0:00:12.94
      ANACOM-MIB::anaBUCAlarms = STRING: "CLEAR"
```

In the following example, snmpset is used to turn TX OFF. Note that the community string “public” will not work when setting a parameter. It will have to be set to something else using the command **SNMP COMMUNITY**. In the this example, it was set to “demoprivate” using the AnaCom, Inc. NMS Supervisor, before running this command from the command line.

```
$ snmpset -v 1 -c demoprivate 192.168.1.44 ANACOM-MIB::anaBUCTxEnable s "OFF"
ANACOM-MIB::anaBUCTxEnable = STRING: "OFF"
```

Note: setting the community string is necessary for using snmpset, but “public” will always work for retrieving data using snmpget.

Note: The AnaCom, Inc. firmware SNMP agent does not support the use of snmptable.