

SNMP Settings and Compliance

Help Guide

Precidia products support SNMPv2c. SNMP (Simple Network Management Protocol) is a protocol for viewing network statistics and settings. An SNMP implementation consists of a manager and an agent that use UDP as a communication protocol.

The SNMP manager is part of the network management realm while the SNMP agent is located in the Precidia unit. The Precidia agent supports MIB-II, and allows GET (read) commands for gathering information, SET (write) commands for managing how the information is gathered, and trap (notification) generation for alerting the Network Manager when selected or unusual events occur. The Precidia SNMP agent uses Community Names to limit access to the MIB variables. Traps, when executed, are sent to specified SNMP managers for review by qualified personnel.

Choose **SNMP Settings** from the System Settings sub-menu.

The SNMP Settings sub-menu appears, as shown below. The SNMP Settings sub-menu has two main sections; Community Names and Trap Managers.

```

Precidia                      CellDial Configuration                      v3.00.00
-----
Device Settings:              SNMP Settings:
1) PPP Dial-Up:                direct | Community Names:
2) Message Handling:           Generic | A) MIBII:                (not set)
                               | B) Precidia:             (not set)
                               | C) Set:                  (not set)
3) Port A - Modem:             disabled |
4) Port B - Network:           disabled |
5) Port C - Network:           disabled |
6) Port D - Network:           disabled |
7) Port E - Network:           disabled |
8) Line Monitor:               disabled |
                               | G) SNMP Trap Settings
*) Save Current Configuration |
-) Exit Configuration (no save) |
$) Security Settings          |
#) System Settings            |
?) Refresh this Screen        |
                               | H) SNMP Allowed Hosts

```

Sample SNMP Settings Sub-menu

Community Names

The Community Names section allows access to the MIBs available in the Precidia unit. The first two sub-menu options allow you to create community names for read-only access to MIBII and Precidia's custom MIB. The MIBII Community Name allows you to use the GET, GETNEXT, and GETBULK commands on supported MIBII variables. Refer to *SNMP Compliance* on page 6 for the MIB variables supported by Precidia.

The MIBII and Precidia Community Names default to “public” if left unset; otherwise, the specified Community Name must be used. The Precidia community name allows you to use the GET, GETNEXT, and GETBULK commands on supported MIBII and Precidia MIB variables. Creating a Community Name for Set allows write access through the SET command to any supported variable that can be set in either the MIBII or Precidia variables. For security purposes, Community Names are displayed as (hidden), however, you can easily check the Community Names by selecting one of the selecting the Community Name menu item. The name will be in brackets.

NOTE: *Precidia does not support the following groups in MIBII:*

AT Group - deprecated
EGP Group
Transmission Group

Creating Community Names

To create or change a community name:

- 1** Choose **System Settings** from the Device Settings menu.
- 2** Choose **SNMP Settings** from the System Settings sub-menu.
- 3** Choose the Community Name type (**MIBII**, **Precidia**, or **Set**) from the SNMP Settings sub-menu.

You are prompted to enter the Community Name with a maximum of 8 characters.

- 4** Type the Community Name at the prompt and press **Enter**.

The Community Name appears as a series of asterisks as it is typed in. On the SNMP Settings sub-menu, (hidden) appears when a Community Name has been set.

Viewing or Deleting Community Names

To view or delete a Community Name:

- 1 Choose the Community Name type (**MIBII**, **Precidia**, or **Set**) from the SNMP Settings sub-menu.

The Community Name appears in brackets at the prompt, like this: [Name].

- 2 **Do not** type any characters. Press **ESC** to return to the menu, or **Enter** to delete the Community Name.

Trap Managers

The Trap Managers section allows you to select two addresses to send reports to, and to set the interval at which the traps are checked. It also contains SNMP Trap Settings, which allows you to set the parameters for detecting and reporting abnormal events, and SNMP Allowed Hosts for limiting the devices that the Precidia unit can accept SNMP commands from.

The Trap Manager is the recipient of any messages generated by the execution of an SNMP trap.

Setting the Trap Manager

To set or change a Trap Manager's IP Address:

- 1 Choose either **IP Address #1**, or **IP Address #2** from the SNMP Settings sub-menu.

You are prompted to enter the IP address of the chosen Manager.

- 2 Type the IP Address at the prompt and press **Enter**.

Checking SNMP traps

To change how often the traps are checked:

- 1 Choose **Trap Check Interval** from the SNMP Settings sub-menu.

You are prompted to enter the number of seconds between Trigger Checking.

- 2 Type the number of seconds for the interval at the prompt and press Enter.

Note that entering 0 disables the trap check.

Setting SNMP traps

There are three parts to setting an SNMP trap:

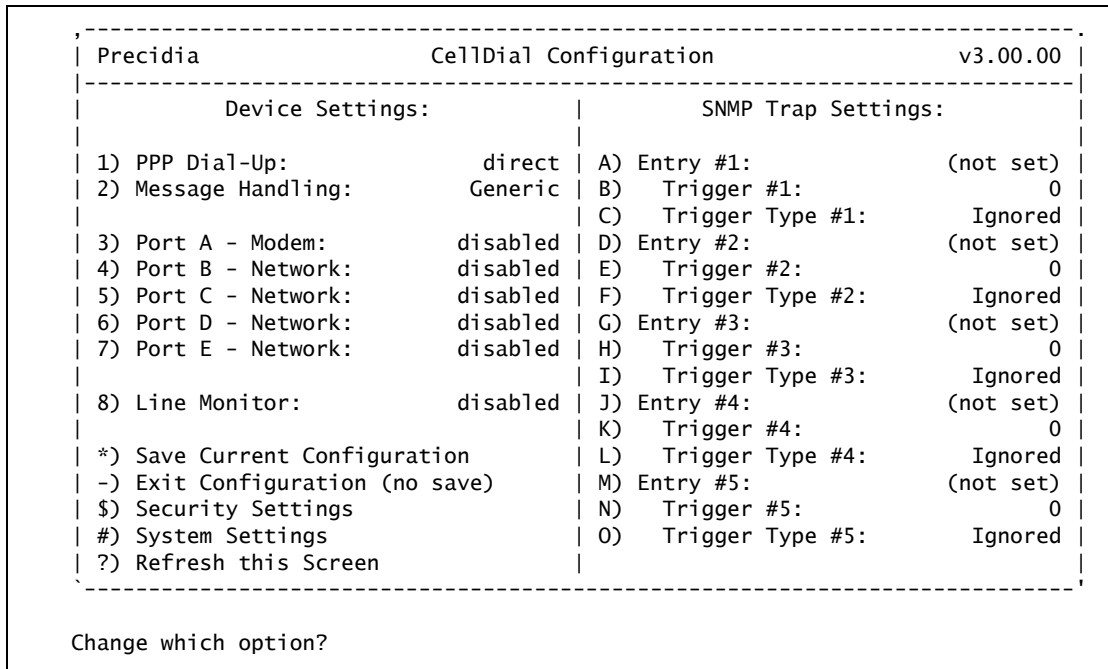
- Entry value, the MIB variable trap to monitor

- Trigger value, the number of times the trigger occurs before notifying the Trap Manager
- Trigger type

To set or change the SNMP traps:

1 Choose **SNMP Trap Settings** from the SNMP Settings sub-menu.

The SNMP Trap Settings sub-menu appears, as shown below.



SNMP Trap Settings Sub-menu

2 Choose an **Entry #** menu item.

You are given a list of entry values and prompted to choose an entry value to monitor. See Table D-1 for descriptions of each entry value.

3 Choose the appropriate entry value from the list and press **Enter**.

4 Choose a **Trigger #** menu item (to define the number of times the trigger occurs before notifying the Trap Manager).

You are prompted to enter a trigger value for the relevant Entry #.

5 Type the trigger value and press **Enter**.

6 Choose a **Trigger Type #** menu item.

You are prompted to select a trigger type, as shown below.

```

A. Ignore Trigger - no traps occur
B. Above Trigger - trap when trigger value exceeded.
C. Below Trigger - trap when value falls below trigger.
D. Delta Value - trap on changes greater than trigger value since
                  last trap check interval.

Choose a trigger type (letter) or press ESC to cancel:
    
```

Selecting the Trigger Type

7 Choose the Trigger Type and press **Enter**.

NOTE: *The Delta option is the most useful and is the Trigger Type normally used.*

Setting allowed hosts

This option allows you to define the IP addresses from which the Precidia unit will accept SNMP commands.

To set or change an Allowed Host's IP Address:

1 Choose **SNMP Allowed Hosts** from the SNMP Settings sub-menu.

The SNMP Hosts Settings sub-menu appears, as shown below.

```

-----
Precidia                      CellDial Configuration                      v3.00.00
-----
Device Settings:              SNMP Host Settings:
1) PPP Dial-Up:                direct | Get/Set Allowed Hosts:
2) Message Handling:           Generic | A) IP Address #1:          0.0.0.0
3) Port A - Modem:             disabled | B) IP Address #2:          0.0.0.0
4) Port B - Network:           disabled | C) IP Address #3:          0.0.0.0
5) Port C - Network:           disabled | D) IP Address #4:          0.0.0.0
6) Port D - Network:           disabled | E) IP Address #5:          0.0.0.0
7) Port E - Network:           disabled | If any address is entered above,
8) Line Monitor:               disabled | SNMP access will be restricted to
*) Save Current Configuration | IP addresses listed above and to
-) Exit Configuration (no save) | the defined SNMP Trap Managers.
$) Security Settings
#) System Settings
?) Refresh this Screen
-----

Change which option?
    
```

SNMP Hosts Settings Sub-menu

2 Choose an **IP Address #** menu item.

You are prompted to enter the IP address of the relevant Allowed Host.

3 Type the IP Address at the prompt and press **Enter**.

NOTE: To restrict access to the Precidia unit you must configure at least one Allowed Host IP (up to five can be configured). You can also turn off SNMP access in the Security Settings sub-menu.

SNMP Compliance

The following tables list the SNMP (Simple Network Management Protocol) variables and commands that Precidia products support. SNMP is an Internet Standard mainly defined in RFC1157, available at www.rfc-editor.org.

- [Table 1: System Group Variables](#)
- [Table 2: Interface Group Variables](#)
- [Table 3: IP Group Variables](#)
- [Table 4: ICMP Group Variables](#)
- [Table 5: TCP Group Variables](#)
- [Table 6: UDP Group Variables](#)
- [Table 7: SNMP Group Variables](#)
- [Table 8: Precidia SNMP Traps](#)

Command columns hold a “Y” if the command is supported, an “N” if the command is not supported, or “n/a” if the command does not apply to the variable.

Table 1: System Group Variables

| Variable Name | Variable Description (Type) | SET | GET | GET-NEXT | GET-BULK |
|---------------|---|-----|-----|----------|----------|
| sysDescr | Textual description of entity with full name and versions of h/w and s/w. (DisplayString) | n/a | Y | Y | Y |
| sysObjectID | Vendor's authoritative ID of network management subsystem. (Object Identifier) | n/a | Y | Y | Y |
| sysUpTime | Time since last re-initialization of agent. (TimeTicks) | n/a | Y | Y | Y |
| sysContact | Contact information for this node (i.e. name of person and contact info). (DisplayString) | Y | Y | Y | Y |
| sysName | Node's fully qualified domain name. (DisplayString) | Y | Y | Y | Y |

Table 1: System Group Variables

| Variable Name | Variable Description (Type) | SET | GET | GET-NEXT | GET-BULK |
|------------------|--|-----|-----|----------|----------|
| sysLocation | Physical location of node. (DisplayString) | Y | Y | Y | Y |
| sysServices | Value indicating set of services supported by this entity. (INTEGER 0...127) | n/a | Y | Y | Y |
| sysOR-LastChange | Value of sysUpTime at time of most recent change in sysORTable. (TimeStamp) | N | N | N | N |
| sysORID | Authoritative identification of an agent capabilities statement. (INTEGER) | N | N | N | N |
| sysORDescr | Textual description of capabilities indentified by sysORID. (DisplayString) | N | N | N | N |
| sysORUpTime | Value of sysUpTime when this row last instantiated. (TimeStamp) | N | N | N | N |

Table 2: Interface Group Variables

| Variable Name | Variable Definition (Type) | SET | GET | GET-NEXT | GET-BULK |
|-------------------|--|-----|-----|----------|----------|
| ifNumber | Number of network i/fs present on system. current operational state of i/f (Integer32) | n/a | Y | Y | Y |
| ifTableLastChange | Time of last create/delete of entry. (Time-Ticks) | N | N | N | N |
| ifIndex | See above. (Integer32 0... 2147483647) | n/a | Y | Y | Y |
| ifDescr | Textual string w/ name of manufacturer etc. (DisplayString) | n/a | Y | Y | Y |
| ifType | I/f type. (IANAifType) | n/a | Y | Y | Y |
| ifMtu | Size of largest datagram able to be sent/ received in octets. (Integer32) | n/a | Y | Y | Y |
| ifSpeed | Estimate of the i/f current bandwidth in bits per second. (Gauge32) | n/a | Y | Y | Y |
| ifPhysAddress | I/f address immediately below the network layer in the protocol stack. (PhysAddress) | n/a | Y | Y | Y |
| ifAdminStatus | Desired state of i/f. (INTEGER) | n/a | Y | Y | Y |

Table 2: Interface Group Variables

| Variable Name | Variable Definition (Type) | SET | GET | GET-NEXT | GET-BULK |
|-------------------|--|-----|-----|----------|----------|
| ifOperStatus | Current operational state of i/f. (INTEGER) | N | N | N | N |
| ifLastChange | Value of sysUpTime at the time i/f entered it's current operational state. (TimeTicks) | n/a | Y | Y | Y |
| ifInOctets | Total number of octets received on i/f including framing characters. (Counter32) | N | N | N | N |
| ifInUcastPkts | Number of subnetwork-unicast packets delivered to higher layers. (Counter32) | n/a | Y | Y | Y |
| ifInDiscards | Number of packets discarded even though no errors (i.e., lack of buffer space). (Counter32) | N | N | N | N |
| ifInErrors | Number of inbound packets with errors preventing delivery to higher layers. (Counter32) | N | N | N | N |
| ifInUnknownProtos | Number of received packets discarded due to unknown/unsupported protocol. (Counter32) | N | N | N | N |
| ifOutOctets | Total octets transmitted out the i/f including framing chars. (Counter32) | N | N | N | N |
| ifOutUcastPkts | Total packets requested to be sent to a subnetwork-unicast address by higher layers. (Counter32) | n/a | Y | Y | Y |
| ifOutDiscards | Number of packets discarded even though no errors. (Counter32) | N | N | N | N |
| ifOutErrors | Number of outbound packets not transmitted due to errors. (Counter32) | N | N | N | N |
| ifOutQLen | Length of output packet queue in packets. (Gauge32) | N | N | N | N |
| ifSpecific | Set to a recognizable oid. (Object Identifier) | N | N | N | N |

Table 3: IP Group Variables

| Variable Name | Variable Description (Type) | SET | GET | GET-NEXT | GET-BULK |
|----------------------|---|------------|------------|-----------------|-----------------|
| ipForwarding | Indication of host or gateway (s/w). (INTEGER) | N | Y | Y | Y |
| ipDefaultTTL | Default value for TTL if not supplied by transport layer. (Integer 1...255) | N | Y | Y | Y |
| ipInReceives | Total number of datagrams received including received in error. (Counter32) | n/a | Y | Y | Y |
| ipInHdrErrors | Number of datagrams discarded due to IP header errors. (Counter32) | n/a | Y | Y | Y |
| ipInAddrErrors | Number of datagrams discarded due to IP address not valid at this location. (Counter32) | n/a | Y | Y | Y |
| ipForwDatagrams | Number of datagrams this not the final destination. (INTEGER) | N | N | N | N |
| ipInUnknownProtos | Number datagrams received successfully but discarded due to unknown/unsupported protocol. (Counter32) | n/a | Y | Y | Y |
| ipInDiscards | Number of datagrams discarded not due to receive problems like lack of buffer space. (INTEGER) | N | N | N | N |
| ipInDelivers | Number of datagrams successfully delivered to IP user-protocols (includes ICMP). (Counter32) | n/a | Y | Y | Y |
| ipOutRequests | Number of datagrams sent by local IP user-protocols (not counted in ipForwDatagrams). (Counter32) | n/a | Y | Y | Y |

Table 3: IP Group Variables

| Variable Name | Variable Description (Type) | SET | GET | GET-NEXT | GET-BULK |
|----------------------|--|------------|------------|-----------------|-----------------|
| ipOutDiscards | Number of datagrams discarded not due to transmission problems (e.g., lack of buffer space). (Counter32) | n/a | Y | Y | Y |
| ipOutNoRoutes | Number of datagrams discarded due to no route found to transmit them. (Counter32) | n/a | Y | Y | Y |
| ipReasmTimeout | Max. number of seconds received fragments are held before reassembly. (Integer32) | N | N | N | N |
| ipReasmReqds | Number of fragments received that needed reassembly. (Counter32) | N | N | N | N |
| ipReasmOKs | Number of successfully reassembled datagrams. (Counter32) | N | N | N | N |
| ipReasmFails | Number of failures detected by IP reassembly algorithm for whatever reason. (Counter32) | N | N | N | N |
| ipFragOKs | Count of successful fragmentations. Not supported. (Counter32) | N | N | N | N |
| ipFragFails | Count of datagrams discarded due to need to frag (i.e. Don't Fragment flag set). (Counter32) | N | N | N | N |
| ipFragCreates | Not supported. (Counter32) | N | N | N | N |
| ipAdEntAddr | IP address. (Octet string) | N | N | N | N |
| ipAdEntIfIndex | Index uniquely identifying the i/f to which this entry is applicable. (INTEGER) | N | N | N | N |
| ipAdEntNetMask | Subnet mask associated with IP address of this entry. (Octet string) | N | N | N | N |
| ipAdEntBcastAddr | Value of the least significant bit in the IP broadcast address. (INTEGER) | N | N | N | N |

Table 3: IP Group Variables

| Variable Name | Variable Description (Type) | SET | GET | GET-NEXT | GET-BULK |
|-------------------------|---|------------|------------|-----------------|-----------------|
| ipAdEntReasmMaxSize | Size of largest datagram that can be reassembled on this i/f. (INTEGER) | N | N | N | N |
| ipNetToMediaIfIndex | Index into table. (INTEGER) | N | N | N | N |
| ipNetToMediaPhysAddress | Physical address. (INTEGER) | N | N | N | N |
| ipNetToMediaNetAddress | IP address corresponding to physical address. (INTEGER) | N | N | N | N |
| ipNetToMediaType | Type of mapping. (INTEGER) | N | N | N | N |
| ipRoutingDiscards | Number of routing entries discarded even though valid. (Counter32) | N | N | N | N |

Table 4: ICMP Group Variables

| Variable Name | Variable Description (Type) | SET | GET | GET-NEXT | GET-BULK |
|----------------------|--|------------|------------|-----------------|-----------------|
| icmpInMsgs | Number of messages received including errors. (Counter32) | n/a | Y | Y | Y |
| icmpInErrors | Number of messages received with CIMP spec. errors. (Counter32) | n/a | Y | Y | Y |
| icmpInDestUnreachs | Number of Destination Unreachable messages received. (Counter32) | N | N | N | N |
| icmpInTimeExcds | Number of Time Exceeded messages received. (Counter32) | N | N | N | N |
| icmpInParmProbs | Number of Parameter Problems messages received. (Counter32) | N | N | N | N |
| icmpInSrcQuenchs | Number of Source Quench messages received. (Counter32) | N | N | N | N |
| icmpInRedirects | Number of Redirect messages received. (Counter32) | N | N | N | N |

Table 4: ICMP Group Variables

| Variable Name | Variable Description (Type) | SET | GET | GET-NEXT | GET-BULK |
|----------------------|--|------------|------------|-----------------|-----------------|
| icmpInEchos | Number of Echo (request) messages received. (Counter32) | n/a | Y | Y | Y |
| icmpInEchoReps | Number of Echo Reply messages received. (Counter32) | N | N | N | N |
| icmpInTimestamps | Number of Timestamp (request) messages received. (Counter32) | N | N | N | N |
| icmpInTimestampReps | Number of Timestamp Reply messages received. (Counter32) | N | N | N | N |
| icmpInAddrMasks | Number of Address Mask Request messages received. (Counter32) | N | N | N | N |
| icmpInAddrMaskReps | Number of Address Mask Replies received. (Counter32) | N | N | N | N |
| icmpOutMsgs | Number of messages attempted to be sent (including errors). (Counter32) | n/a | Y | Y | Y |
| icmpOutErrors | Number of ICMP messages not sent due to internal error (e.g. buffer is full) (Counter32) | N | N | N | N |
| icmpOutDestUnreachs | Number of Destination Unreachable messages sent. (Counter32) | N | N | N | N |
| icmpOutTimeExcds | Number of Time Exceeded messages sent. (Counter32) | N | N | N | N |
| icmpOutParmProbs | Number of Parameter Problem messages sent. (Counter32) | N | N | N | N |
| icmpOutSrcQuenchs | Number of Source Quench messages sent. (Counter32) | N | N | N | N |
| icmpOutRedirects | Number of Redirect messages sent (0 for host). (Counter32) | N | N | N | N |
| icmpOutEchos | Number of Echo (requests) messages sent. (Counter32) | N | N | N | N |
| icmpOutEchoReps | Number of Echo Reply messages sent. (Counter32) | n/a | Y | Y | Y |

Table 4: ICMP Group Variables

| Variable Name | Variable Description (Type) | SET | GET | GET-NEXT | GET-BULK |
|-----------------------|--|-----|-----|----------|----------|
| icmpOutTimestamps | Number of Timestamp (request) messages sent. (Counter32) | N | N | N | N |
| icmpOutTimestampsReps | Number of Timestamp Reply messages sent. (Counter32) | N | N | N | N |
| icmpOutAddrMasks | Number of Address Mask Requests sent.(Counter32) | N | N | N | N |
| icmpOutAddrMaskReps | Number of Address Mask Replies sent. (Counter32) | N | N | N | N |

Table 5: TCP Group Variables

| Variable Name | Variable Definition (Type) | SET | GET | GET-NEXT | GET-BULK |
|-----------------|---|-----|-----|----------|----------|
| tcpRtoAlgorithm | Algorithm used (defined in MIBII). (INTEGER) | n/a | Y | Y | Y |
| tcpRtoMin | Retransmission timeout min in milliseconds. (Integer32) | n/a | Y | Y | Y |
| tcpRtoMax | Retransmission timeout max in milliseconds. (INTEGER) | N | N | N | N |
| tcpMaxConn | Number of TCP conns able to be supported. (Integer32) | n/a | Y | Y | Y |
| tcpActiveOpens | Number of times transition from CLOSED to SYN-SENT. (Counter32) | n/a | Y | Y | Y |
| tcpPassiveOpens | Number of times transition from SYN-RCVD to LISTEN. (Counter32) | n/a | Y | Y | Y |
| tcpAttemptFails | Number of times transition from SYN-SENT or SYN-RCVD to CLOSED and from SYN-RCVD to LISTEN. (Counter32) | n/a | Y | Y | Y |
| tcpEstabResets | Number of transitions from ESTAB or CLOSE-WAIT to CLOSED. (Counter32) | n/a | Y | Y | Y |

Table 5: TCP Group Variables

| Variable Name | Variable Definition (Type) | SET | GET | GET-NEXT | GET-BULK |
|----------------------|---|------------|------------|-----------------|-----------------|
| tcpCurrEstab | S/W - total conns in ESTAB or CLOSE-WAIT. (Gauge32) | n/a | Y | Y | Y |
| tcpInSegs | Total segments received, including error segments. (Counter32) | n/a | Y | Y | Y |
| tcpOutSegs | Total transmitted segments, not including retransmitted segments. (Counter32) | n/a | Y | Y | Y |
| tcpRetransSegs | Total retransmitted segments. (Counter32) | N | N | N | N |
| tcpConnState | The state of the connection. (INTEGER) | N | Y | Y | Y |
| tcpConnLocalAddress | Local IP address. (Octet string) | n/a | Y | Y | Y |
| tcpConnLocalPort | Local port number. (INTEGER) | n/a | Y | Y | Y |
| tcpConnRemoteAddress | Remote IP address. (Octet string) | n/a | Y | Y | Y |
| tcpConnRemotePort | Remote port number. (INTEGER) | n/a | Y | Y | Y |
| tcpInErrs | Segments received with errors (e.g., bad checksum). (Counter32) | n/a | Y | Y | Y |
| tcpOutRsts | Segments sent w/ RST. (Counter32) | n/a | Y | Y | Y |

Table 6: UDP Group Variables

| Variable Name | Variable Definition (Type) | SET | GET | GET-NEXT | GET-BULK |
|----------------------|--|------------|------------|-----------------|-----------------|
| udpInDatagrams | Total packets received. (Counter32) | n/a | Y | Y | Y |
| udpNoPorts | Packets with no applicatgion at port (i.e. no socket). (Counter32) | n/a | Y | Y | Y |

Table 6: UDP Group Variables

| Variable Name | Variable Definition (Type) | SET | GET | GET-NEXT | GET-BULK |
|-----------------|--|-----|-----|----------|----------|
| udpInErrors | Packets undeliverable other than no application at port (i.e., errors, full buffer). (Counter32) | n/a | Y | Y | Y |
| udpOutDatagrams | Total packets sent. (Counter32) | n/a | Y | Y | Y |
| udpLocalAddress | Local IP Address. (Octet string) | n/a | Y | Y | Y |
| udpLocalPort | Local port number. (INTEGER 0...65535) | n/a | Y | Y | Y |

Table 7: SNMP Group Variables

| Variable Name | Variable Description (Type) | SET | GET | GET-NEXT | GET-BULK |
|-------------------------|---|-----|-----|----------|----------|
| snmpInPkts | Total number of messages delivered to SNMP agent. (Counter32) | n/a | Y | Y | Y |
| snmpOutPkts | Total number of SNMP messages delivered to agent. (Counter32) | n/a | Y | Y | Y |
| snmpInBadVersions | Total number of SNMP messages delivered to agent for unsupported SNMP version. (Counter32) | n/a | Y | Y | Y |
| snmpInBadCommunityNames | Total number of messages using unknown SNMP community name. (Counter32) | n/a | Y | Y | Y |
| snmpInBadCommunityUses | Total messages received requesting operation not supported by community named in message. (Counter32) | n/a | Y | Y | Y |
| snmpInASNParseErrs | Total ASN.1 or BER errors. (Counter32) | N | N | N | N |
| snmpEnableAuthenTraps | Indicates whether the SNMP agent is permitted to generate authentication-failure traps. (INTEGER - 1,2) | N | N | N | N |

Table 7: SNMP Group Variables

| Variable Name | Variable Description (Type) | SET | GET | GET-NEXT | GET-BULK |
|-----------------------|---|------------|------------|-----------------|-----------------|
| snmpSilentDrops | Total number of PDUs (Packet Data Units) silently dropped because the size of the reply was greater than a local constraint or the max message size of the originator of the request. (Counter32) | n/a | Y | Y | Y |
| snmpProxyDrops | Total number of PDUs silently dropped because the transmission of the message to a proxy target failed. (Counter32) | N | N | N | N |
| coldStart | Signifies that the SNMPv2 entity, acting in an agent role, is reinitializing itself and that its configuration may have been altered. | N | N | N | N |
| warmStart | Signifies that the SNMPv2 entity, acting in an agent role, is reinitializing itself such that its configuration is unaltered. | N | N | N | N |
| linkUp | Signifies that the sending protocol entity recognizes that one of the communication links represented in the agent's configuration has come up. | N | N | N | N |
| authenticationFailure | Signifies that the SNMPv2 entity, acting in an agent role, has received a protocol message that is not properly authenticated. While all implementations of the SNMPv2 must be capable of generating this trap, the snmpEnableAuthenTraps object indicates whether this trap will be generated. | N | N | N | N |

The Precidia SNMP agent provides a method for enabling traps (notifications), but it is not in strict accordance with the SNMP protocol. Table 8 lists the traps the Precidia SNMP agent supports.

Table 8: SNMP Traps

| MIB Variable | Description |
|---------------------|--|
| ipInReceives | Total number of datagrams received including received in error |
| ipInAddrErrors | Number of datagrams discarded due to IP address not valid at this location |
| ipInUnknownProto | Number datagrams received successfully but discarded due to unknown/unsupported protocol |
| ipInDelivers | Number of datagrams successfully delivered to IP user-protocols (includes ICMP) |
| ipOutRequests | Number of datagrams sent by local IP user-protocols (not counted in ipForwDatagrams) |
| ipOutNoRoutes | Number of datagrams discarded due to no route found to transmit them |
| icmpInMsgs | Number of messages received including errors |
| icmpInEchos | Number of Echo (request) messages received |
| icmpOutMsgs | Number of messages attempted to be sent (including errors) |
| tcpActiveOpens | Number of transitions from CLOSED to SYN-SENT |
| tcpPassiveOpens | Number of transitions from SYN-RCVD to LISTEN |
| tcpEstabResets | Number of transitions from ESTAB or CLOSE-WAIT to CLOSED |
| tcpCurrEstab | Total connections in ESTAB or CLOSE-WAIT |
| tcpInSegs | Total segments received, including error segments |
| tcpOutSegs | Total retransmitted segments |
| tcpOutRsts | Segments sent with RST |
| udpInDtgrms | Total packets received (udpInDatagrams) |
| udpNoPorts | Packets with no application at port |
| udpOutDatagrams | Total packets sent |
| routerHostInNoRoute | Total number of messages inbound from hosts with no device found. |
| routerHostOutErrors | Total number of messages outbound to hosts not sent due to errors. |
| routerHostTimeouts | Total number of messages from hosts not sent due to timeout. |

Table 8: SNMP Traps

| MIB Variable | Description |
|---------------------|---|
| routerHostDiscards | Total number of messages from hosts that were discarded. |
| routerHostReversals | Total number of messages from hosts that were sent back to the host. |
| routerDevOutErrors | Total number of messages outbound to devices not sent due to errors. |
| routerDevDiscards | Total number of messages from devices that were discarded. |
| snmpInPkts | Total number of messages delivered to SNMP agent |
| snmpOutPkts | Total number of packets delivered by SNMP agent |
| snmpInBadComName | Total number of messages using unknown SNMP community name (snmpInBadCommunityNames) |
| snmpInBadComUses | Total messages received requesting operation not supported by community named in message (snmpInBadCommunityUses) |