AKCP

Est. USA 1981

www.AKCP.com

Using AKCP OID's & SNMP commands

Table of Contents

Introduction

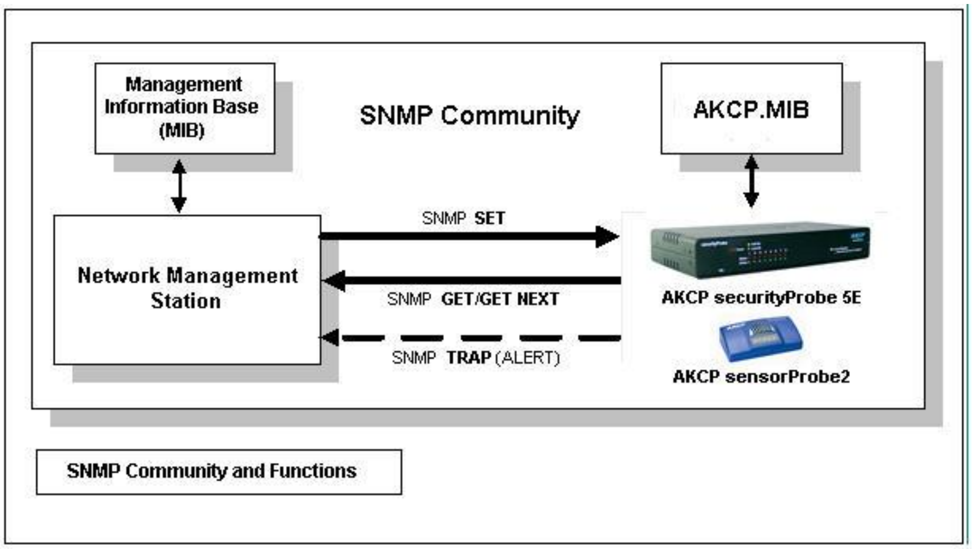
What is an OID?

An OID is an “object identifier”. In computer networking the OID is used to identify an object that is stored in a Management Information Base (MIB). The MIB is a database that manages the devices used in a network setup. In this case, the OID’s are used to identify the objects, such as sensors, that are connected to your unit. Roughly speaking, each OID identifies a variable that can be read or set via an SNMP command.

What is SNMP?

SNMP is the most popular network management protocol in the TCP/IP protocol suite.
It is a simple request/response protocol that communicates management information between two types of SNMP software entities: SNMP applications (also called SNMP managers) and SNMP agents (AKCP base units and sensors).

SNMP applications run in a network management station (NMS) and issue queries to gather information about the status, configuration, and performance of the AKCP devices and sensors. For example, this very simple diagram belows shows this concept of how WhatsUp Gold software, a network management station, and an AKCP securityProbe 5ES, sensorProbe2 or sensorProbe+ communicate via SNMP accross the TCP\IP network:



A list of all of the options that are available for use from a command can be printed by passing "-h" on the command line to any of the commands.

How to use this manual

This manual aims to provide the user with a guide on how to use SNMP commands and the AKCP OID’s. It does assume you have previous knowledge through completion of previous tutorials on our products

If you have any difficulty viewing the screen shots or the text in this manual, you can use the “Zoom In” feature of your document reader.

Please also note the ***“Quick Tips”*** in each section and also the ***“Quick Facts & FAQ”*** at the end of this manual for more information on SNMP commands and OID’s.

Nagios and the sensorProbe Expansion Modules

You can download the check_sprobe version for Nagios here:

http://www.akcp.com/ftpuser/Tools/check_sprobe

```
Usage: check_sprobe -H host [-C community] -T type -p port -P udp_port
       check_sprobe --help
       check_sprobe --version
```

```
Example: check_sprobe -H 192.168.0.100 -C public -T Temperature -p 1 -P 10161
Example: check_sprobe -H 192.168.0.100 -C public -I 1 -P 10161
```

AKCP OIDs

Important note: *if a sensor is in error, there is no SNMP response for its OID requesting its value.*

Each sensor can be individually addressed by SNMP.
The OID is composed of the main OID number which is completed by the index (or compound ID) of the sensor. The index is fixed and will be the same on all units.
The index represents the position of the sensor, the last 3 being: module number, port number and subport number.

sensorProbe OIDs

For the sensorProbe units, all of the AKCP intelligent sensors are divided into three groups, as listed below. All of the sensors in the same group have the same OID.
The securityProbe and sensorProbe+ units support a wider range of sensors and therefore have more sensor groups with different OIDs. You'll need to use the MIB browser or the unit's Web UI to find the correct OID. An example is shown for SP+ units at the end of this manual.

The **sensorProbe OID groups** are as follows:

1. The first group includes temperature sensors only:

SP
sensorProbeTempDegree = 1.3.6.1.4.1.3854.1.2.2.1.16.1.3.X
sensorProbeTempStatus = 1.3.6.1.4.1.3854.1.2.2.1.16.1.4.X

2. The second group includes all analog sensors such as humidity, airflow, 4-20mA, and DC Voltage, excluding the temperature sensor:

SP
sensorProbeHumidityPercent = 1.3.6.1.4.1.3854.1.2.2.1.17.1.3.X
sensorProbeHumidityStatus   = 1.3.6.1.4.1.3854.1.2.2.1.17.1.4.X

3. The third group includes all switch-type sensors such as water sensor, dry contact, security, motion sensor, AC Voltage Detector, relay, and siren & strobe light.

SP
sensorProbeSwitchStatus = 1.3.6.1.4.1.3854.1.2.2.1.18.1.3.X

Where: X is a port number starting from 0 to 7 for SP8, or 0 to 27 for SP8X20.
So for example an SP OID value that looks like this:
.1.3.6.1.4.1.3854.1.2.2.1.16.1.3.0
Would be the value for a temperature sensor status connected to RJ45 port 1.

Note: *The dual temperature and humidity sensor consists of a temperature sensor* ***(group1)*** *and a humidity sensor* ***(group2).***

Temperature Sensor OID's (sensorProbe only):
.1.3.6.1.4.1.3854.1.2.2.1.16.1.3.0 #this is the temperature 1
.1.3.6.1.4.1.3854.1.2.2.1.16.1.3.1 #this is the temperature 2
.1.3.6.1.4.1.3854.1.2.2.1.16.1.3.2 #this is the temperature 3
.1.3.6.1.4.1.3854.1.2.2.1.16.1.3.3 #this is the temperature 4
.1.3.6.1.4.1.3854.1.2.2.1.16.1.3.4 #this is the temperature 5
.1.3.6.1.4.1.3854.1.2.2.1.16.1.3.5 #this is the temperature 6
.1.3.6.1.4.1.3854.1.2.2.1.16.1.3.6 #this is the temperature 7
.1.3.6.1.4.1.3854.1.2.2.1.16.1.3.7 #this is the temperature 8

This shows the OID value for a temperature sensor connected to each port. As you can see the final digit changes from 0-7 showing the 8 ports of an SP8 unit.

As noted earlier, the securityProbe and sensorProbe+ units support a wider range of sensors and therefore have more sensor groups with different OIDs. You’ll need to use the MIB browser or the unit’s Web UI to find the correct OID. An example is shown for SP+ units at the end of this manual.

Switch Type sensors (sensorProbe group 3):

OID’s can be used to set a switch output to high. For example, if you wished to remotely activate a device then you could use an OID. The status of a switch (high/low) can be found using the following OID (sensorProbe only):

```
.1.3.6.1.4.1.3854.1.2.2.1.18.1.3.x
```

Where “x” is the value (0-7 on 8 port unit) that the switch is connected.

If you wish to set the status of the switch to high then you can use the following OID (sensorProbe only):

```
.1.3.6.1.4.1.3854.1.2.2.1.18.1.8.x
```

Again the “x” value is the port to which the switch is connected.

This can now be used for remote activation of a device. If a temperature sensor, for example, is connected on port 1 and a dry contact cable is connected on port 2, and the temperature exceeds the threshold, then the management software can send an SNMP Set to the dry contact on port 2 with the following command:

```
snmpset <ipaddress> <password> .1.3.6.1.4.1.3854.1.2.2.1.18.1.8.1  i  1
```

The i denotes an integer value
i can take 2 values: 1 = output high, 2 = output low
If set to high it will make the dry contact pins output +5V which can turn on a relay.

<u>Quick Tip:</u>

If you want to control an output of a Dry contact, DO NOT use online/offline to switch the dry contact on and off. Leave the Dry contact output online and use 'sensorProbeSwitchOutputLevel OID' instead.

```
snmpset -v1 -c <password> <ipaddress> .1.3.6.1.4.1.3854.1.2.2.1.18.1.8.0 i 0
```

This OID will tell port 1 to set to 0 Volts/Close

```
snmpset -v1 -c <password> <ipaddress> .1.3.6.1.4.1.3854.1.2.2.1.18.1.8.0 i 1
```

This OID will tell port 1 to set to +5 Volts/Open

Relay and Siren OIDs (sensorProbe only)

We have special SNMP OIDs for controlling relays and sirens. These are:

sensorProbeSwitchManualRelayAction = .1.3.6.1.4.1.3854.1.2.2.1.18.1.25.x
Where: x is a port number from 0 to 7 for SensorProbe (SP2, SP8, SP8L and CP8)

The integer values of this OID are:
1 = Allow sensor to control
3 = Turn On
4 = Turn Off
7 = Cycle Off-On-Off
8 = Cycle On-Off-On

The SNMP OID for controlling relays is the following:
.1.3.6.1.4.1.3854.1.2.2.1.18.1.25.0

Integer Value :
1 = allow-sensor-control
3 = relay-turnon
4 = relay-turnoff
7 = cycle Off-On-Off
8 = cycle On-Off-On

Dry Contact status (sensorProbe only)

Here is SNMP OID for reading the dry contact status: .1.3.6.1.4.1.3854.1.2.2.1.18.1.3.(Port-1)

The return value can be:
noStatus(1),
normal(2),
highCritical(4),
lowCritical(6),
sensorError(7),

sensorProbeSwitchStatus = 1.3.6.1.4.1.3854.1.2.2.1.18.1.3.X

Where: X is a port number starting from 0 to 7 for SP8, or 0 to 27 for SP8X20.

OIDs for SEC 5ES and Expansion Modules

This SNMPwalk command .1.3.6.1.4.1.3854.2.3.1.1 can also be used for checking all of the sensors on the securityProbe 5ES.

The OID's for **checking the status of the sensors on the unit** are as follows:

> These are the OID's which detect if there’s a warning/critical on a particular module:
> .1.3.6.1.4.1.3854.1.2.2.1.117.1.0 get number of sensor that the status is not normal
> .1.3.6.1.4.1.3854.1.2.2.1.117.2.0 get number of sensor status is Critical and Error
> .1.3.6.1.4.1.3854.1.2.2.1.117.3.0 get number of sensor status is Error

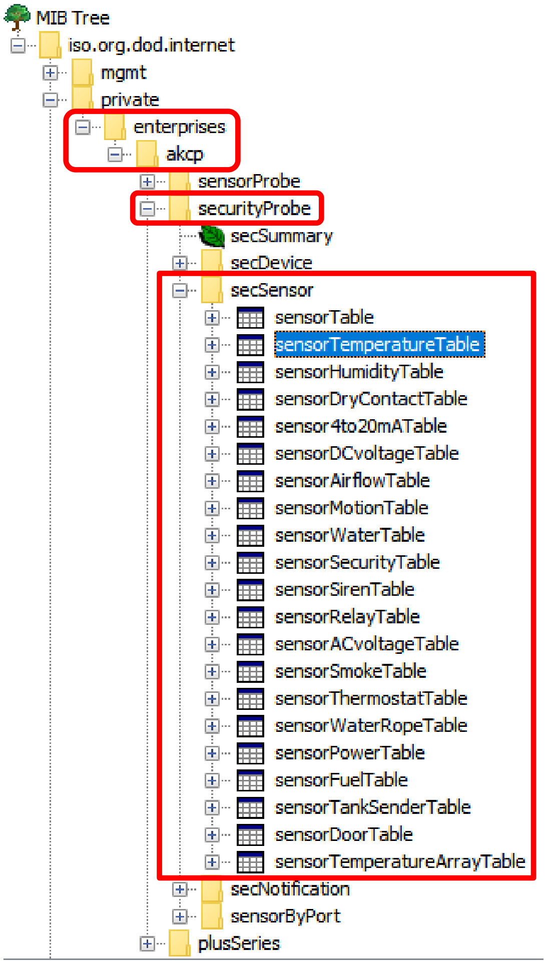
The OID's for **Virtual Analog Sensors** on the securityProbe 5ES are:

> Current Reading
> .1.3.6.1.4.1.3854.1.2.2.1.19.30.2.1.13.<port>
>
> Unit Text
> .1.3.6.1.4.1.3854.1.2.2.1.19.30.2.1.27.<port>

In recent firmware the securityProbe has been updated to use the **sensorByPort branch OIDs**. This will allow you to keep using the same OIDs for multiple units, which is very useful for mass deployments (using configuration backups). The compound ID format is used, similarly as on the sensorProbe+ units. This index is dependent on the sensor location.

The SEC compound ID is as following:
expansionPort, boardPosition, sensorPort, sensorPosition

See below for an example.



- MIB Tree
  - iso.org.dod.internet
    - mgmt
    - private
      - enterprises
        - akcp
          - sensorProbe
          - securityProbe
            - secSummary
            - secDevice
            - secSensor
              - sensorTable
              - sensorTemperatureTable
              - sensorHumidityTable
              - sensorDryContactTable
              - sensor4to20mATable
              - sensorDCvoltageTable
              - sensorAirflowTable
              - sensorMotionTable
              - sensorWaterTable
              - sensorSecurityTable
              - sensorSirenTable
              - sensorRelayTable
              - sensorACvoltageTable
              - sensorSmokeTable
              - sensorThermostatTable
              - sensorWaterRopeTable
              - sensorPowerTable
              - sensorFuelTable
              - sensorTankSenderTable
              - sensorDoorTable
              - sensorTemperatureArrayTable
            - secNotification
            - sensorByPort
          - plusSeries

You have to use the OIDs from the securityProbe series, in the secSensor subtree, each sensor type as a corresponding table of OIDs that can be used.
To address a sensor property, you need to use the corresponding OID + the sensor index, like with **sensorTemperatureDegree.0.0.1.1** where sensorTemperatureDegree corresponds to the OID (.1.3.6.1.4.1.3854.2.3.2.1.4) and .0.0.1.1 is the sensor index.

You can either use the unit's Web UI (the *Get SNMP OID button*) or the MIB browser (see an example at the end of this manual) to find your correct OIDs.

Expansion unit OIDs on securityProbe

You can check the OID's of each sensor connected to the expansion units by using this snmpwalk command:

```
snmpwalk -v1 -c<community> <ipaddress>
.1.3.6.1.4.1.3854.2.3.1.1.2
```

This will show the list of all the sensors descriptions on unit.
The expansion unit will keep the sensor with the sensor ID. The last number of the result of each OID will be the sensor ID that you can use for getting the other values.

You can use this command below for **getting the sensor readings on expansion units**:

```
snmpwalk -v1 -c<community> <ipaddress> .1.3.6.1.4.1.3854.2.3.1.1.4
```

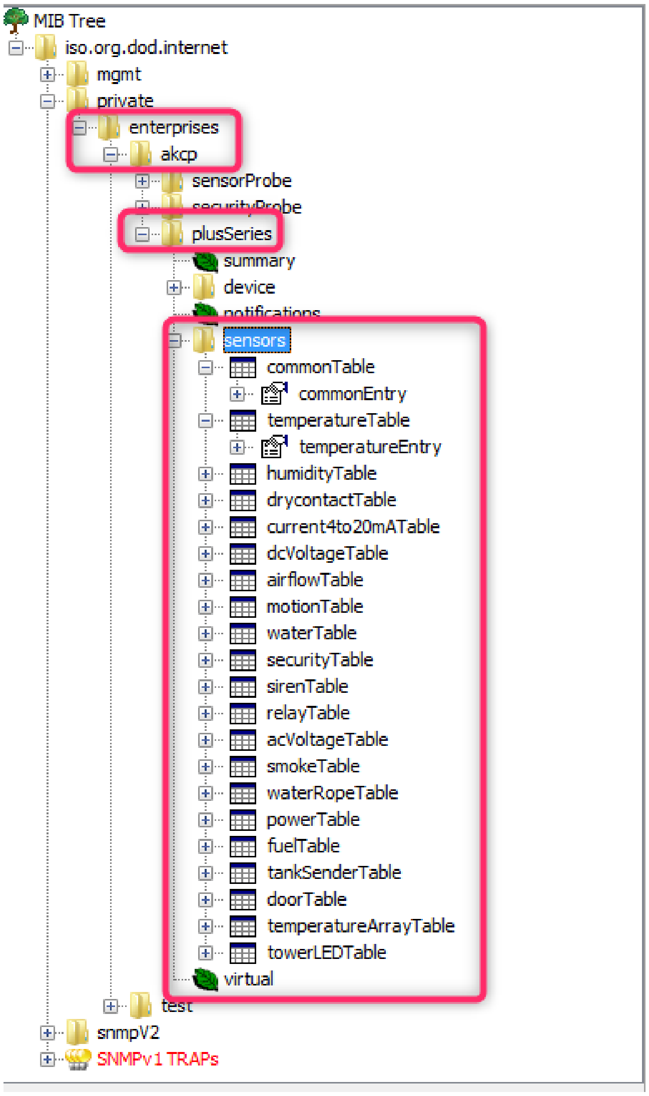
OIDs for SP+ units

The OID architecture is a little different on the SP+ family than on the SEC & SP2/8 family.

The OIDs for the SP+ family is composed of 2 blocks:

1: the OID: as described in the MIB file, for example: .akcp.plusSeries.commonTable.commonEntry.commonDescription (.1.3.6.1.4.1.3854.3.5.1.1.2)

2: the sensor index (compound ID): this index is dependent on the sensor location as following:
expansionPort (COM port), boardPosition (Board), personalityCardPosition (Module), sensorPort, sensorPosition (Subport)
for example: 0.0.0.3.3 - on SP2+ is internal com, main board, main module, RJ45 port 4, subport 4



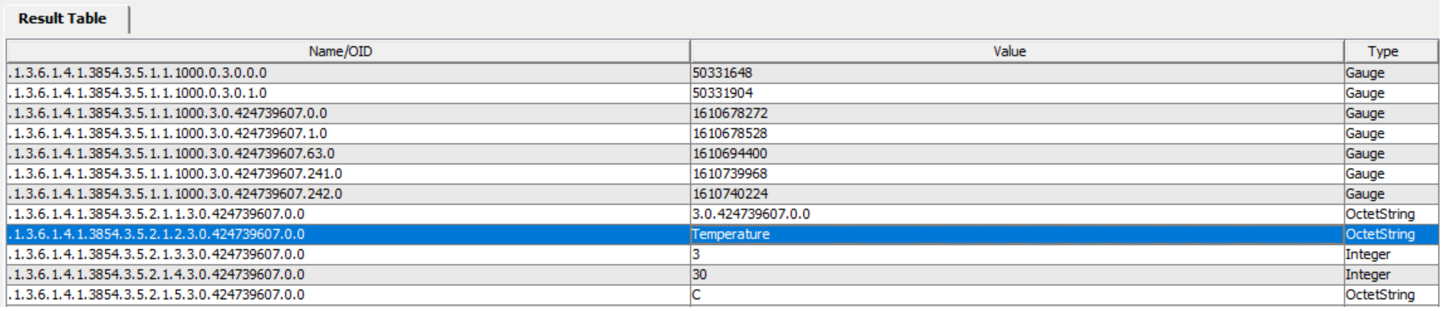
You have to use the OIDs from the SP+ series (plusSeries), in the sensors subtree, each sensor type as a corresponding table of OIDs that can be used. To address a sensor property, you need to use the corresponding OID + the sensor index, like with **temperatureDegree.0.0.1.0.1** where temperatureDegree corresponds to the OID (.1.3.6.1.4.1.3854.3.5.2.1.4) and .0.0.1.0.1 is the sensor index.

Therefore if you use SEC/SP8 OIDs to poll the SP+ unit this will not work, the SP+ uses a different branch of OIDs in the MIB file that starts with .1.3.6.1.4.1.3854.3 and called plusSeries.

You can either use the unit’s Web UI (the *Get SNMP OID button*) or the MIB browser (see an example at the end of this manual) to find your correct OIDs.

OIDs for Wireless Sensors

The OID architecture of the wireless sensors are similar to the SP+ family, and contains a compound ID.
The MIB file only contains the static part of the OID, since each and every wireless sensor have a unique network address, and it is not possible to include them in the MIB. You would either need to calculate the OID using the details below, run SNMP Walk or use the “Get SNMP OID” button on the device’s WebUI to find the necessary OID.



Result Table

Name/OID	Value	Type
.1.3.6.1.4.1.3854.3.5.1.1.1000.0.3.0.0.0	50331648	Gauge
.1.3.6.1.4.1.3854.3.5.1.1.1000.0.3.0.1.0	50331904	Gauge
.1.3.6.1.4.1.3854.3.5.1.1.1000.3.0.424739607.0.0	1610678272	Gauge
.1.3.6.1.4.1.3854.3.5.1.1.1000.3.0.424739607.1.0	1610678528	Gauge
.1.3.6.1.4.1.3854.3.5.1.1.1000.3.0.424739607.63.0	1610694400	Gauge
.1.3.6.1.4.1.3854.3.5.1.1.1000.3.0.424739607.241.0	1610739968	Gauge
.1.3.6.1.4.1.3854.3.5.1.1.1000.3.0.424739607.242.0	1610740224	Gauge
.1.3.6.1.4.1.3854.3.5.2.1.1.3.0.424739607.0.0	3.0.424739607.0.0	OctetString
.1.3.6.1.4.1.3854.3.5.2.1.2.3.0.424739607.0.0	Temperature	OctetString
.1.3.6.1.4.1.3854.3.5.2.1.3.3.0.424739607.0.0	3	Integer
.1.3.6.1.4.1.3854.3.5.2.1.4.3.0.424739607.0.0	30	Integer
.1.3.6.1.4.1.3854.3.5.2.1.5.3.0.424739607.0.0	C	OctetString

We will explain the structure of the OID using this BOS LBTH TemperatureDegree OID:
.1.3.6.1.4.1.3854.3.5.2.1.4.3.0.424739607.0.0

1.3.6.1.4.1.3854.3.5.2.1.4.3.0. is a static OID part for all wireless sensors (BOS/WTS).

424739607 is a decimal representation of the wireless sensor’s network address - you would need to convert the address HEX to decimal to get this value.

To get this value take the Device Network Address (Hex) from the sensor, ex.: 19510317
Convert the HEX value to Decimal (using Windows Calculator for example): 424739607

0.0 is the compound index from the wireless sensor’s internal mapping.
It will be different for all wireless sensors, in this format: 3.0.<NetworkAddress>.<Port>.<Subport>

For example, for a WTTH sensor, the last 2 number of the OID is referencing to:
0.0 for temperature
1.0 for humidity

We will list the compound index value below for all currently released wireless sensors (as of 2021 April).

Wireless Sensor Compound Index Values

BOS Sensors

BOS LBTH:

63.0: MCU voltage
0.0: Temperature
1.0: Humidity

BOS LBDC:

63.0: MCU voltage	
1.0: Dry contact input	(port 1)
1.1: Edge counter	(port 1)
2.0: Dry contact input	(port 2)
2.1: Edge counter	(port 2)
3.0: Dry contact input	(port 3)
3.1: Edge counter	(port 3)
4.0: Dry contact input	(port 4)
4.1: Edge counter	(port 4)
5.0: Dry contact input	(port 5)
5.1: Edge counter	(port 5)

BOS LBAD:

63.0: MCU voltage	
1.0: Current 4..20 mA	(port 1)
2.0: Voltage 0..5 V	(port 2)
3.0: Current 4..20 mA	(port 3)
4.0: Voltage 0..5 V	(port 4)

BOS LBSW:

63.0: MCU voltage
0.0: Dry contact input
0.1: Edge counter

BOS LBPC:

63.0: MCU voltage
0.0: Dry contact input
0.1: Pulse counter
1.0: Flow

BOS LBTD:
63.0: MCU voltage
0.0: Temperature
1.0: Humidity
2.0: Dry contact input
2.1: Edge counter

BOS LBTL:
63.0: MCU voltage
0.0: Fluid height
1.0: Fluid volume

BOS LBTM:
63.0: MCU voltage
0.0: Temperature front top
0.1: Temperature front middle
0.2: Temperature front bottom
0.3: Humidity front middle
1.0: Temperature difference top
1.1: Temperature difference middle
1.2: Temperature difference bottom
2.0: Temperature rear top
2.1: Temperature rear middle
2.2: Temperature rear bottom
2.3: Humidity rear middle

BOS LBWD:
63.0: MCU voltage
0.0: Water Detector
0.1: Edge counter

BOS LSSI:
63.0: MCU voltage
0.0: LoRa SNR
1.0: LoRa RSSI

BOS LDPTHM:

63.0: MCU voltage
0.0: Temperature front top
0.1: Temperature front middle
0.2: Temperature front bottom
0.3: Humidity front middle
1.0: Temperature difference top
1.1: Temperature difference middle
1.2: Temperature difference bottom
2.0: Temperature rear top
2.1: Temperature rear middle
2.2: Temperature rear bottom
2.3: Humidity rear middle
3.0: Differential pressure
4.0: Dry contact input (port 1)
4.1: Edge counter (port 1)
5.0: Dry contact input (port 2)
5.1: Edge counter (port 2)

BOS LDC10:

63.0: MCU voltage
0.0: Dry contact input (port 1)
0.1: Edge counter (port 1)
1.0: Dry contact input (port 2)
1.1: Edge counter (port 2)
2.0: Dry contact input (port 3)
2.1: Edge counter (port 3)
3.0: Dry contact input (port 4)
3.1: Edge counter (port 4)
4.0: Dry contact input (port 5)
4.1: Edge counter (port 5)
5.0: Dry contact input (port 6)
5.1: Edge counter (port 6)
6.0: Dry contact input (port 7)
6.1: Edge counter (port 7)
7.0: Dry contact input (port 8)
7.1: Edge counter (port 8)
8.0: Dry contact input (port 9)
8.1: Edge counter (port 9)
9.0: Dry contact input (port 10)
9.1: Edge counter (port 10)

BOS LBDP:

63.0: MCU voltage
0.0: Differential pressure

BOS LBMB:

63.0: MCU voltage
1.0: Modbus analog (port 1)
2.0: Modbus analog (port 2)
3.0: Modbus analog (port 3)
4.0: Modbus analog (port 4)
5.0: Modbus analog (port 5)
6.0: Modbus analog (port 6)
7.0: Modbus analog (port 7)
8.0: Modbus analog (port 8)
9.0: Modbus analog (port 9)
10.0: Modbus analog (port 10)
11.0: Modbus analog (port 11)
12.0: Modbus analog (port 12)
13.0: Modbus switch (port 1)
14.0: Modbus switch (port 2)
15.0: Modbus switch (port 3)
16.0: Modbus switch (port 4)
17.0: Modbus counter (port 1)
18.0: Modbus counter (port 2)
19.0: Modbus counter (port 3)
20.0: Modbus counter (port 4)

WTS Sensors

WTS WTTH:

63.0: Battery voltage
0.0: Temperature
1.0: Humidity

WTS WTDC:

63.0: Battery voltage	
1.0: Dry contact input	(port 1)
1.1: Edge counter	(port 1)
2.0: Dry contact input	(port 2)
2.1: Edge counter	(port 2)
3.0: Dry contact input	(port 3)
3.1: Edge counter	(port 3)
4.0: Dry contact input	(port 4)
4.1: Edge counter	(port 4)
5.0: Dry contact input	(port 5)
5.1: Edge counter	(port 5)

WTS WTAD:

63.0: Battery voltage	
1.0: Current 4..20 mA	(port 1)
2.0: Voltage 0..5 V	(port 2)
3.0: Current 4..20 mA	(port 3)
4.0: Voltage 0..5 V	(port 4)

WTS WTSW:

63.0: Battery voltage
0.0: Dry contact input
0.1: Edge counter

WTS WTPC:

63.0: Battery voltage
0.0: Dry contact input
0.1: Pulse counter
1.0: Flow

WTS WTPT:

63.0: Battery voltage
0.0: Temperature (RTD PT100)

WTS WTTD:

63.0: Battery voltage
0.0: Temperature
1.0: Humidity
2.0: Dry contact input
2.1: Edge counter

WTS WTTL:

63.0: Battery voltage
0.0: Fluid height
1.0: Volume

WTS WTTM:

63.0: Battery voltage
0.0: Temperature front top
0.1: Temperature front middle
0.2: Temperature front bottom
0.3: Humidity front middle
1.0: Temperature difference top
1.1: Temperature difference middle
1.2: Temperature difference bottom
2.0: Temperature rear top
2.1: Temperature rear middle
2.2: Temperature rear bottom
2.3: Humidity rear middle

WTS WTWD:

63.0: Battery voltage
0.0: Water Detector
0.1: Edge counter

WTS WSSI:

63.0: Battery voltage
0.0: LoRa SNR
1.0: LoRa RSSI

WTS WTDP:

63.0: Battery voltage
0.0: Differential pressure

WTS WTMB:
63.0: Battery voltage
1.0: Modbus analog (port 1)
2.0: Modbus analog (port 2)
3.0: Modbus analog (port 3)
4.0: Modbus analog (port 4)
5.0: Modbus analog (port 5)
6.0: Modbus analog (port 6)
7.0: Modbus analog (port 7)
8.0: Modbus analog (port 8)
9.0: Modbus analog (port 9)
10.0: Modbus analog (port 10)
11.0: Modbus analog (port 11)
12.0: Modbus analog (port 12)
13.0: Modbus switch (port 1)
14.0: Modbus switch (port 2)
15.0: Modbus switch (port 3)
16.0: Modbus switch (port 4)
17.0: Modbus counter (port 1)
18.0: Modbus counter (port 2)
19.0: Modbus counter (port 3)
20.0: Modbus counter (port 4)

WTS WTPR:
63.0: Battery voltage
0.0: Temperature
1.0: Pressure

WTS WTLL:
63.0: Battery voltage
0.0: Temperature
1.0: Fluid height
2.0: Fluid volume

WTS WTTN:
63.0: Battery voltage
0.0: Temperature
1.0: Operable sensors count
2.0: Calibration status

WTS WTHN:

63.0: Battery voltage
0.0: Temperature
1.0: Humidity
2.0: Operable sensors count
3.0: Calibration status

WTS WTAQ:

63.0: Battery voltage
0.0: Temperature
1.0: Humidity
2.1: Particle number concentration PM0.5
3.0: Particle mass concentration PM1.0
3.1: Particle number concentration PM1.0
4.0: Particle mass concentration PM2.5
4.1: Particle number concentration PM2.5
5.0: Particle mass concentration PM4.0
5.1: Particle number concentration PM4.0
6.0: Particle mass concentration PM10
6.1: Particle number concentration PM10
7.0: Typical particle size
8.0: Air quality VOC index

WTS WTIO8:

63.0: Battery voltage

1.0: Dry contact input/output	(port 1)
1.1: Edge counter	(port 1)
2.0: Dry contact input/output	(port 2)
2.1: Edge counter	(port 2)
3.0: Dry contact input/output	(port 3)
3.1: Edge counter	(port 3)
4.0: Dry contact input/output	(port 4)
4.1: Edge counter	(port 4)
5.0: Dry contact input/output	(port 5)
5.1: Edge counter	(port 5)
6.0: Dry contact input/output	(port 6)
6.1: Edge counter	(port 6)
7.0: Dry contact input/output	(port 7)
7.1: Edge counter	(port 7)
8.0: Dry contact input/output	(port 8)
8.1: Edge counter	(port 8)

Quick Facts and FAQ

Question:
What is the OID for the 8 port relay box on the sensorProbe?

Answer:
.1.3.6.1.4.1.3854.1.2.2.1.19.29.1.2.<port>.25.<subport>

Question:
In an NMS (like PRTG or Cacti) I need to fill an OID to get a graph. Which OID do I have to use for this? How can I translate the output of the snmpwalk command to a usable OID for graphing?

Answer:
On the sensorProbe units you can run an SNMPwalk of:
Humidity:
iso.3.6.1.4.1.3854.2.3.3.1.2.2 --> this gives the description
iso.3.6.1.4.1.3854.2.3.3.1.4.2 --> this is the OID that belongs to the description above

Temperature:
iso.3.6.1.4.1.3854.2.3.2.1.2.3 --> this give the description
iso.3.6.1.4.1.3854.2.3.2.1.20.3 --> this is the OID that belongs to the description above

Question:
Do you have an OID for the securityProbe and securityProbe 5ES that will tell me if any of my sensors connected to any of the RJ-45 sensor ports is in an alarm state?

Answer:
Yes, we do. You can use this OID here: .1.3.6.1.4.1.3854.1.1.2.0

This OID will return the worst case status of the SEC or SEC 5ES. So, for example you have a sensor in a normal, a warning state and a sensor in a critical state, the OID will return that a sensor on the unit that is in the critical state.

From here you can do an SNMPwalk on all the OID’s to determine which sensors are in what status. This can be utilized by say a NMS for example.

Question:

Do you have an OID that will return what the MAC ID of my unit is on the sensorProbe units?

Answer:

Yes, we do. You can use this OID here: 1.3.6.1.4.1.3854.1.2.2.1.3.0

Question:

I have used the web interface to configure the sensorProbe device. In particular, I have used the web interface to set temperature thresholds; can these thresholds be set via SNMP commands?

Answer:

Yes, you can use the following SNMP set commands for changing your temperature\humidity thresholds for sensorProbe units.

Temperature

```
snmpset -m all -v 1 -c <community> <IPAddress> .1.3.6.1.4.1.3854.1.2.2.1.16.1.7.<port> i <High Warning value>

snmpset -m all -v 1 -c <community> <IPAddress> .1.3.6.1.4.1.3854.1.2.2.1.16.1.8.<port> i <High Critical value>

snmpset -m all -v 1 -c <community> <IPAddress> .1.3.6.1.4.1.3854.1.2.2.1.16.1.9.<port> i <Low Warning value>
snmpset -m all -v 1 -c <community> <IPAddress> .1.3.6.1.4.1.3854.1.2.2.1.16.1.10.<port> i <Low Critical value>
```

Humidity

```
snmpset -m all -v 1 -c <community> <IPAddress> .1.3.6.1.4.1.3854.1.2.2.1.17.1.7.<port> i <High Warning value>
snmpset -m all -v 1 -c <community> <IPAddress> .1.3.6.1.4.1.3854.1.2.2.1.17.1.8.<port> i <High Critical value>
snmpset -m all -v 1 -c <community> <IPAddress> .1.3.6.1.4.1.3854.1.2.2.1.17.1.9.<port> i <Low Warning value>
snmpset -m all -v 1 -c <community> <IPAddress> .1.3.6.1.4.1.3854.1.2.2.1.17.1.10.<port> i <Low Critical value>
```

Question:

What is the OID for the daisyTemp temperature string sensor on sensorProbe?

Answer:

You can use the following SNMP Set commands for the daisyTemp sensor on SP units.

For the daisyTemp sensors and Relay Array the port starts at "1" thresholds.

For checking it the SNMP command is:

```
snmpget -m all -v1 -c <community> <IPaddress> .1.3.6.1.4.1.3854.1.2.2.1.101.0
```

For setting it the SNMP command is:

```
snmpget -m all -v1 -c <community> <IPaddress> .1.3.6.1.4.1.3854.1.2.2.1.101.0 i <start port>
```

Question:

Do you have an OID for checking the sensor type, or what type of sensor is connected to my sensorProbe base unit?

Answer:

Yes, you can use the SensorType OID for checking the sensor type on each port and also use the Sensor Online OID again for making sure that sensor is online.

.1.3.6.1.4.1.3854.1.2.2.1.18.1.9 for checking the type of sensor

When polling this for all sensors on your unit you will get this displayed:

```
1.3.6.1.4.1.3854.1.2.2.1.18.1.9.0 8  <- airflow
1.3.6.1.4.1.3854.1.2.2.1.18.1.9.1     9  <- siren & strobe
1.3.6.1.4.1.3854.1.2.2.1.18.1.9.2     3  <- ?
1.3.6.1.4.1.3854.1.2.2.1.18.1.9.3     3  <- ?
1.3.6.1.4.1.3854.1.2.2.1.18.1.9.4     3  <- ?
1.3.6.1.4.1.3854.1.2.2.1.18.1.9.5     13 <- relay / voltage
1.3.6.1.4.1.3854.1.2.2.1.18.1.9.6     3  <- ?
1.3.6.1.4.1.3854.1.2.2.1.18.1.9.7     3  <- ?
```

The values represent the sensors as follows:

temperature(1),
4-20mA(2),
humidity(3),
water(4),
atod(5),

security(6),
airflow(8),
siren(9),
dryContact(10),
voltage(12),
relay(13),
motion(14)

Sensor controlled relay:

```
snmpset -mall -v1 -c<set community> <ip> .1.3.6.1.4.1.3854.1.2.2.1.18.1.7 i <stage>
```

when
<set community> is admin password
<ip> is IPaddress of unit
<stage> is 0 for close, 1 for open

What is the MIB Browser?

A MIB Browser is a tool that allows you to pull out data from SNMP enabled devices such as routers, switches, and servers. In addition to pulling out data, a MIB Browser may also perform the following functions:

- Retrieve and display MIB data in a human readable form
- Allows you to make it smarter by compiling in additional MIB definitions
- Set MIB variables, create, modify or delete conceptual table rows
- Basic alarm (trap) management
- View the MIB text file in a graphical manner usually featuring a MIB tree
- Support all versions of SNMP including SNMPv3 security

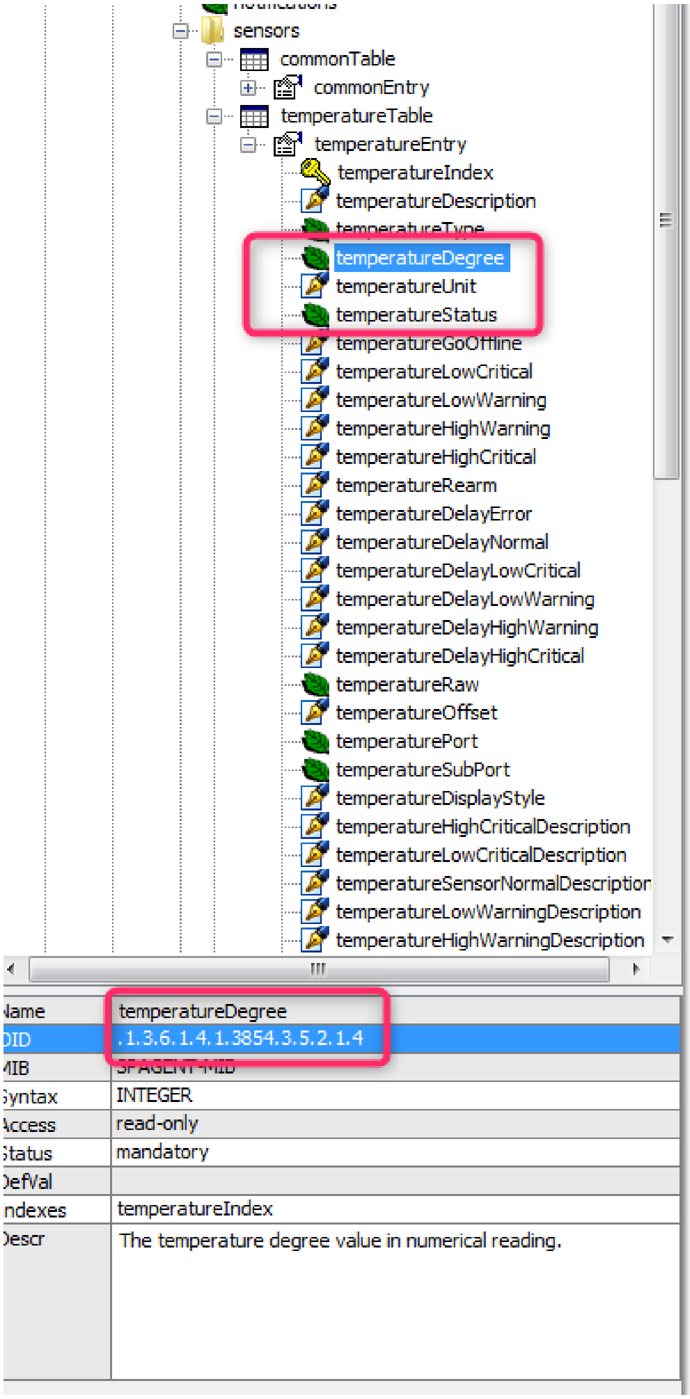
You can download this tool for free from iReasoning’s web site here:
http://ireasoning.com/mibbrowser.shtml

We’ll show an example use of this tool in the next section.

Example finding OIDs for SP+

After opening the MIB browser, first load the AKCP.MIB file using the File menu / Load MIBs option.

Then you can navigate the tree view (per product family) and find all base OIDs from the MIB browser navigation tree, as in the example screenshot below for SP+ temperature sensors:



lame	temperatureDegree
DID	.1.3.6.1.4.1.3854.3.5.2.1.4
MIB	SPAGENT-MIB
Syntax	INTEGER
Access	read-only
Status	mandatory
DefVal	
ndexes	temperatureIndex
Descr	The temperature degree value in numerical reading.

You will find the following OIDs:

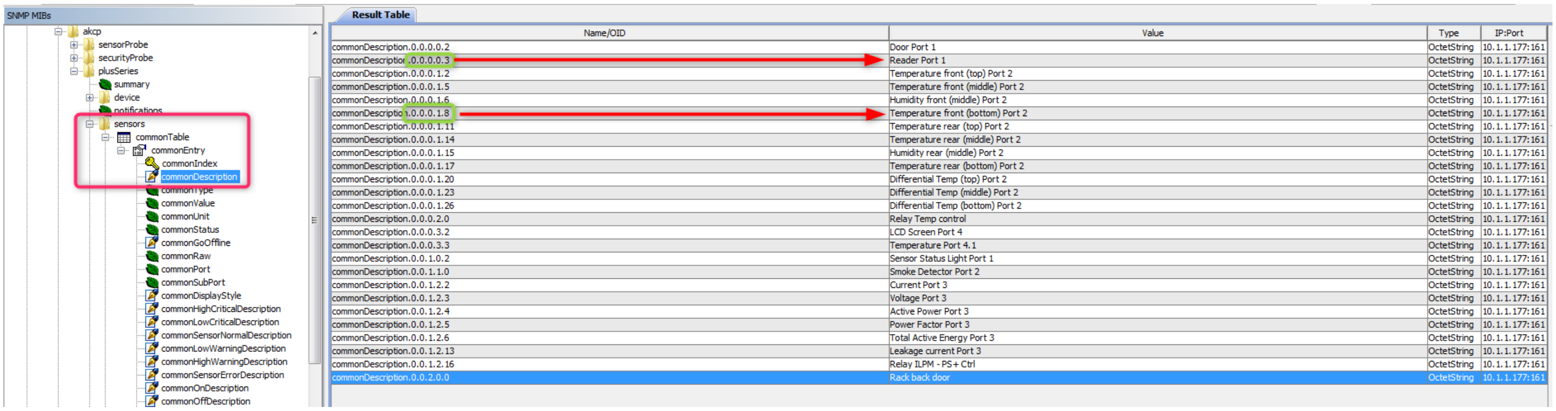
temperatureDegree : .1.3.6.1.4.1.3854.3.5.2.1.4
temperatureUnit : .1.3.6.1.4.1.3854.3.5.2.1.5
tempertureStatus : .1.3.6.1.4.1.3854.3.5.2.1.6

humidityPercen: .1.3.6.1.4.1.3854.3.5.3.1.4
humidityUnit : .1.3.6.1.4.1.3854.3.5.3.1.5
humidityStatus : .1.3.6.1.4.1.3854.3.5.3.1.6

Then, you have to add the sensor Index, which depends on the sensor position (module, port and its subport), in our example 0.0.1.0.1 means: module 1, port 1, subport 2.

The subport index is for multiple sensors connected to a physical port. For example, the THS sensor has 2 sensors (temperature and humidity) connected to one port, their subport would be respectively 1 and 0.

Using the MIB browser, you can quickly get the sensor indexes by double-clicking on the commonDescription OID in the navigation tree. This will display the descriptions of all sensors connected to your SP+, and their corresponding OID+index as in the screenshot below:



Name/OID	Value	Type	IP:Port
commonDescription.0.0.0.0.2	Door Port 1	OctetString	10.1.1.177:161
commonDescription.0.0.0.0.3	Reader Port 1	OctetString	10.1.1.177:161
commonDescription.0.0.0.1.2	Temperature front (top) Port 2	OctetString	10.1.1.177:161
commonDescription.0.0.0.1.5	Temperature front (middle) Port 2	OctetString	10.1.1.177:161
commonDescription.0.0.0.1.6	Humidity front (middle) Port 2	OctetString	10.1.1.177:161
commonDescription.0.0.0.1.8	Temperature front (bottom) Port 2	OctetString	10.1.1.177:161
commonDescription.0.0.0.1.11	Temperature rear (top) Port 2	OctetString	10.1.1.177:161
commonDescription.0.0.0.1.14	Temperature rear (middle) Port 2	OctetString	10.1.1.177:161
commonDescription.0.0.0.1.15	Humidity rear (middle) Port 2	OctetString	10.1.1.177:161
commonDescription.0.0.0.1.17	Temperature rear (bottom) Port 2	OctetString	10.1.1.177:161
commonDescription.0.0.0.1.20	Differential Temp (top) Port 2	OctetString	10.1.1.177:161
commonDescription.0.0.0.1.23	Differential Temp (middle) Port 2	OctetString	10.1.1.177:161
commonDescription.0.0.0.1.26	Differential Temp (bottom) Port 2	OctetString	10.1.1.177:161
commonDescription.0.0.0.2.0	Relay Temp control	OctetString	10.1.1.177:161
commonDescription.0.0.0.3.2	LCD Screen Port 4	OctetString	10.1.1.177:161
commonDescription.0.0.0.3.3	Temperature Port 4.1	OctetString	10.1.1.177:161
commonDescription.0.0.1.0.2	Sensor Status Light Port 1	OctetString	10.1.1.177:161
commonDescription.0.0.1.1.0	Smoke Detector Port 2	OctetString	10.1.1.177:161
commonDescription.0.0.1.2.2	Current Port 3	OctetString	10.1.1.177:161
commonDescription.0.0.1.2.3	Voltage Port 3	OctetString	10.1.1.177:161
commonDescription.0.0.1.2.4	Active Power Port 3	OctetString	10.1.1.177:161
commonDescription.0.0.1.2.5	Power Factor Port 3	OctetString	10.1.1.177:161
commonDescription.0.0.1.2.6	Total Active Energy Port 3	OctetString	10.1.1.177:161
commonDescription.0.0.1.2.13	Leakage current Port 3	OctetString	10.1.1.177:161
commonDescription.0.0.1.2.16	Relay ILPM - PS+ Ctrl	OctetString	10.1.1.177:161
commonDescription.0.0.2.0.0	Rack back door	OctetString	10.1.1.177:161

Due to the very flexibility of our units, we can't give you directly a specific OID for each sensors connected to your unit as it depends on the units configuration and on the sensor location and type.

That is why we have this architecture with the combination of the OID from the MIB + the sensor Index because every user configuration is different from one to another.

Please contact support@akcp.com if you have any further technical questions or problems.

Thanks for Choosing AKCP!