



Sensorsoft[®] SNMP Agent User Manual

Agent Software Version 2.00

Manual P/N 071-0047 Rev 3 November 11, 2002

Copyright © 2000-2002 Sensorsoft Corporation, All rights reserved.
Sensorsoft, Soft Thermometer, Soft Humidity Meter, Soft Power Sensor, Soft Moisture Sensor,
Soft Relay, SCOM and Remote Watchman are trademarks of Sensorsoft Corporation.

Table of Contents

About This Manual.....	5
Overview	6
Installation.....	7
Configuring the Agent	7
Sensorsoft SNMP Agent Specifications.....	9
General Usage	10
Scalar and Boolean Class Variables and Devices	10
Reading and Manipulating MIB Variables	10
Traps	11
Agent Software Upgrades	11
Description of Variables in the Sensorsoft MIB	12
trapDestination.....	12
agentVersion	12
location	13
status	13
model	13
numberOfScalarVariables.....	13
scalarIndex	13
scalarVarName.....	14
scalarData	14
unitOfMeasure	14
scalarRWCcapability	14
scalarCriticalHighThreshold.....	15
scalarCriticalLowThreshold.....	15
scalarWarningHighThreshold.....	15
scalarWarningLowThreshold.....	15
scalarCriticalTrapInterval	16
scalarWarningTrapInterval	16
sumberOfBooleanVariables.....	16
booleanIndex.....	16
booleanVarName	17
booleanData	17
booleanOneStateString	17
booleanZeroStateString	17
booleanRWCcapability.....	18

booleanCriticalState.....	18
booleanCriticalTrapInterval.....	18
Description of Traps Issued by the Agent.....	19
Troubleshooting Guide.....	20
Troubleshooting Checklist.....	20
Common Problems.....	21
CAUTION NOTICE.....	24
Restarting the Sensorsoft SNMP Agent software.....	25
Stopping the Sensorsoft SNMP Agent software.....	26
Getting Help.....	27
Technical Support.....	27
30 Day No Risk Guarantee.....	27
Returns.....	27

About This Manual

This documentation contains information about installing, configuring, using and troubleshooting the Sensorsoft SNMP Agent. It does *not* contain documentation specific to your specific network management software or tools, nor does it provide documentation for the MSS100 on which the Sensorsoft SNMP Agent software runs. Separate documentation for the MSS100 is included with the unit.

The following conventions are used throughout this manual:

The term Agent refers to the Sensorsoft SNMP Agent software.

When the names of Sensorsoft MIB variables occur they are displayed in **bold**.

When additional sections of this manual are referenced, *Italics* are used.

GET and SET, capitalized, refer to the SNMP mechanisms for retrieving and modifying data on the Sensorsoft SNMP Agent.

The term Sensorsoft device and its acronym SSD refers to devices whose model number appears in the section *Sensorsoft SNMP Agent Specifications* under “Sensorsoft Protocol devices supported”.

The term Network Management System, abbreviated NMS, refers to the SNMP based software system or tools from which the Sensorsoft SNMP Agent is being monitored or controlled.

MSS is used to refer to the Lantronix MSS100 Device Server, where the Sensorsoft SNMP Agent software resides and runs.

Overview

The Sensorsoft SNMP Agent is software that is embedded into the Lantronix MSS100 Device Server hardware. It allows any RS232 Sensorsoft device (SSD) to be accessed over a TCP/IP network from user-supplied SNMP-based software tools or a Network Management Station (NMS). The Sensorsoft device plugs into the MSS100 RS232 serial port. The Lantronix MSS100 plugs into an Ethernet 10BASE-T network.

The Sensorsoft SNMP Agent responds to SNMP v1 GET and SET commands. It can issue traps when user-controlled warning and critical levels are breached, a new Sensorsoft device is plugged in, or communication with an existing Sensorsoft device is lost. The Sensorsoft SNMP Agent uses one private enterprise Sensorsoft MIB to accommodate all current Sensorsoft devices. A MIB description file is provided for compilation by the NMS software to access one Sensorsoft device with up to six variables.

Installation

1. Place the Lantronix MSS100 Device Server in its desired location.
2. Connect the MSS100 to your network using a Sensorsoft P/N C2006 (or equivalent) cable.
3. Make the appropriate power connection using the provided power adapter.
4. Place the Sensorsoft device in an appropriate location, and connect it to the serial port on the back of the MSS100 using a Sensorsoft P/N C2001 cable. The RJ45 connector on the C2001 plugs into the SSD and the DB-25 connector plugs into the serial port on the MSS100.
5. Copy the MIB file into the include directory of your SNMP Network Management System (NMS) software.

Configuring the Agent

1. If the network settings (IP address and subnet mask) of the MSS100 were not custom configured for your network you will now have to do so. Information on how to do this is provided in the Lantronix MSS100 Device Server documentation.
2. Configure your NMS software to communicate with the MSS100 at your configured IP address.
3. Load the **sensorsoft1.mib** file into your NMS software as per the procedure recommended by the manufacturer.
4. Set the timeout feature of your NMS software for at least 6000 milliseconds (6 seconds) to work properly with the Sensorsoft SNMP Agent.
5. Set the community name for this Sensorsoft SNMP Agent to **public** in the NMS software.
6. Once your NMS software can perform GET/SET requests, SET the following MIB variables to their desired values.

For Scalar Class Sensorsoft devices:

```
trapDestination
location
unitOfMeasure
scalarCriticalHighThreshold
scalarCriticalLowThreshold
scalarWarningHighThreshold
scalarWarningLowThreshold
scalarCriticalTrapInterval
scalarWarningTrapInterval
```

For Boolean Class Sensorsoft devices:

```
trapDestination
location
booleanCriticalState
booleanCriticalTrapInterval
```

7. According to the instructions provided by your NMS software, configure it to receive traps from the Sensorsoft SNMP Agent.

It is a good idea to test the system to make sure traps are being received properly. Testing can take various forms, depending on the type of Sensorsoft device available. For example, if testing with a Sensorsoft Thermometer, consider using a blow dryer to raise the temperature above the specified high thresholds. If using a Sensorsoft device whose data cannot easily be manipulated, such as a Sensorsoft Humidity Meter, consider temporarily manipulating the thresholds to values that will trigger traps.

Sensorsoft SNMP Agent Specifications

The following specifications apply to the Sensorsoft SNMP Agent software. For hardware specifications of the Lantronix MSS100, please refer to the MSS100 Installation Guide.

Community names: public only with read and write permissions

SNMP Version supported: v1

Sensorsoft Protocol devices supported: ST6105; ST6154; ST6100; ST6150; SM6204; SM6201; SR6170; SP6400.

Maximum number of Sensorsoft Boolean or Scalar class variables supported: 6

Variable access method: indexed

Trap specific types: criticalHigh(3), criticalLow(4), warningHigh(5), warningLow(6), zeroCritical(1), oneCritical(2), sensorRemoved(7), newSensor(8).

Maximum trap rate: 1 per minute

Number of Sensorsoft MIB variables: 26

Starting Sensorsoft MIB OID: .1.3.6.1.4.1.912.1.2.0

Sensorsoft MIB Variables: *trapDestination, agentVersion, *location, status, model, numberOfScalarVariables, scalarIndex, scalarVarName, scalarData, *unitofMeasure, *scalarRWCcapability, *scalarCriticalHighThreshold, *scalarCriticalLowThreshold, *scalarWarningHighThreshold, *scalarWarningLowThreshold, *scalarCriticalTrapInterval, *scalarWarningTrapInterval, numberOfBooleanVariables, booleanIndex, booleanVarName, booleanData, booleanOneStateString, booleanZeroStateString, booleanRWCcapability, *booleanCriticalState, *booleanCriticalTrapInterval. Variables with an asterisk are saved in non-volatile memory.

Communications Protocol over MSS100 serial port: Sensorsoft Device Protocol, Version 1.0/2.0 (auto switching)

General Usage

The Sensorsoft SNMP Agent software is controlled through a series of variables that can be used to read and manipulate a Sensorsoft device (SSD). (These variables are described in detail under *Description of the Variables in the Sensorsoft MIB*.) The typical way to manipulate the Sensorsoft SNMP Agent's variables is to load the provided MIB into third party, SNMP-enabled software that can act as an interface between the user and the Sensorsoft SNMP Agent.

The administrator typically sets certain parameters on the Sensorsoft SNMP Agent, such as alerting thresholds and a trap destination. The Sensorsoft SNMP Agent then signals important events by sending "traps" to the trap destination (typically to the computer that is running the network management software). The exact procedure for setting the Sensorsoft SNMP Agent's parameters and receiving traps are dependent upon the tools used for network management.

CAUTION: The Sensorsoft SNMP Agent runs on a specially configured Lantronix MSS100 Device Server. Modifying the MSS's settings could cause the Sensorsoft SNMP Agent to stop functioning properly. It is recommended that you do not modify any MSS settings other than the following (see the MSS Installation Guide for their meanings):

IPADDRESS
SUBNET MASK
NAMESERVER
DOMAIN
GATEWAY
PRIVPASS

The MSS provides a mechanism to reset to factory-default settings. If you find that you must reset the MSS to assign an IP address, you will have to follow the procedure later in this Guide to restart the Agent software.

Scalar and Boolean Class Variables and Devices

Sensorsoft devices (SSD) and their variables can be classified in terms of the way they represent their data. Scalar variables are those values that are represented by a range of possible numbers. Devices that monitor temperature or humidity are classified as scalar. Boolean variables define only two states: binary one or zero. Devices that monitor power (PWR FAIL/PWR OK), flooding (WET/DRY), or control a relay position (ON/OFF) are classified as boolean.

The distinction between scalar and boolean is important because it influences the way data is retrieved from the Sensorsoft SNMP Agent. If you examine the MIB, you will notice variables prefixed with *Scalar* or *Boolean*. Only one set of variables is of interest to you, depending on the type of SSD being used. When using a Scalar SSD the values of the variables prefixed with Scalar are used and not those prefixed with Boolean. When using a Boolean SSD the values of the variables prefixed with Boolean are used and not those prefixed with Scalar. See the *Description of Variables in the MIB* section for a more detailed description of each MIB variable.

Reading and Manipulating MIB Variables

Another important distinction to be made is the difference between the Sensorsoft SNMP Agent's variables and the SSD's variables. The SSD's contain the information that is useful to the user.

The user does not have direct access to those variables. Instead they are accessed through the Agent's MIB variables.

Measurements and readings from Scalar and Boolean SSD variables are accessed through the **scalarData** and **booleanData** MIB variables, respectively. Although your SSD could contain multiple variables, the Sensorsoft SNMP Agent allows access to one variable at a time. The variable that is currently being accessed is controlled by an index stored in **scalarIndex** or **booleanIndex**. Setting **scalarIndex** or **booleanIndex** to a new value changes the variable set that is being accessed, so that all subsequent GET and SET operations operate on that variable.

See the *Description of Variables in the MIB* section for a detailed description of each variable. Information about each of the Sensorsoft SNMP Agent's variables is also contained in the MIB, and can be viewed using the MIB browser in many network management software packages. Consult your NMS software's documentation to determine how to do this.

Caution: Some SSDs, such as the SM6204 Sensorsoft Humidity Meter , have a calibration variable (**scalarVarName** = *CAL*, **scalarIndex** = 2). Modifying this variable will likely cause your sensor to display incorrect readings and require recalibration at Sensorsoft Labs.

Traps

Traps are the Sensorsoft SNMP Agent's way of letting the NMS know when something significant occurs at the Agent. The Agent sends traps to the IP address specified in the variable **trapDestination**. Each trap has a number associated with it, allowing the recipient to determine what exactly occurred, and to take appropriate action.

For scalar variables, two severity levels of traps can be issued: warning traps and critical traps. This severity is arbitrary, although the intent is that the NMS will take different action depending on how severe the situation is. A plausible scenario is to have advisories sent in response to warning traps, and to take more drastic action, such as shutting off the power, in response to critical traps. Note that a critical state takes precedence over a warning state. In other words, if a variable breaches a warning and critical threshold, only critical traps will be issued (at the interval specified in **scalarCriticalTrapInterval**).

In the case of Boolean variables, only one type of trap is provided. **booleanCriticalState** traps are sent when the **booleanData** variable is equal to **booleanCriticalState** variable.

Agent Software Upgrades

Software upgrades for the Sensorsoft SNMP Agent, if available, are offered at no charge up to one year from purchase date. After one year an upgrade fee applies. There are two ways to upgrade. One is to return the Lantronix MSS100 unit to Sensorsoft Corporation, shipping and brokerage pre-paid by you, where we will upgrade the Agent software and return it to you for a nominal shipping charge. The other upgrade method can be performed by providing us with access to your Lantronix MSS100 over the Internet. In this case we would require the IP address and system password for your MSS100 unit to perform the Agent software upgrade.

Description of Variables in the Sensorsoft MIB

This section lists and describes all of the variables contained in the Sensorsoft MIB. In addition, to the name and description the following information is given:

<i>Type</i>	The type of variable, describes the values it can hold. Typical variable types include: <i>IpAddress</i> ; <i>Integer</i> ; <i>String</i> . Attempting to enter an invalid value (for example, the string "hello" for a variable that requires a number) will usually result in the change request being discarded. When changing a value, always verify that the change actually occurred.
<i>Access</i>	Specifies the read and write access granted for the variable. It can be <i>Read/Write</i> or <i>Read only</i> . Variables that are <i>Read only</i> indicate that the variable is provided for informational purposes and cannot be set by the user. Do not confuse the Access listed with the read/write permissions of the variables on the SSD itself. Permissions listed in the access field describe the write access for the Agent variables, rather than the SSD's internal variables.
<i>OID</i>	The specific Object Identifier for the variable. This may be required by some SNMP tools to specifically reference the variable. Most NMS software will allow you to refer to the variable name, and will keep track of the OID for you.
<i>Non-Volatile</i>	Variables marked non-volatile are saved between sessions. That is, if the MSS100 loses power (either deliberately or accidentally), the Agent's non-volatile variables will not lose their values.

Note that many network management software packages can extract this information directly from the MIB. Consult your NMS software's documentation for information on how to view the following information from your NMS.

trapDestination

DESCRIPTION: The destination IP address for sending traps.

TYPE: IP address, of the form xxx.xxx.xxx.xxx

ACCESS: Read/Write

OID .1.3.6.1.4.1.912.1.2.0

Non-volatile

agentVersion

DESCRIPTION: A text string describing the name and version number of the agent.

TYPE: String

ACCESS: Read only

OID .1.3.6.1.4.1.912.1.3.0

location

DESCRIPTION: User-defined description of where the Sensorsoft Agent is located. This can be used to easily identify the particular Sensorsoft SNMP Agent.

TYPE: String

ACCESS: Read/Write

OID .1.3.6.1.4.1.912.1.4.0

Non-volatile

status

DESCRIPTION: A string that describes the status of the agent. This can be one of *Normal*, *Offline*, *Critical High*, *Critical Low*, *Warning High*, or *Warning Low*. If both a warning state and critical state are occurring, the critical state takes precedence. An *Offline* indication means that the Agent cannot communicate with the SSD.

TYPE: String

ACCESS: Read only

OID .1.3.6.1.4.1.912.1.5.0

model

DESCRIPTION: The model number and name of the currently connected Sensorsoft device.

TYPE: String

ACCESS: Read only

OID .1.3.6.1.4.1.912.1.6.0

numberOfScalarVariables

DESCRIPTION: The number of scalar variables in the currently connected Sensorsoft device. The specific variable being selected is described by **scalarIndex**.

See also *Scalar and Boolean Class Variables and Devices*.

TYPE: Integer, range 0-6.

ACCESS: Read only

OID .1.3.6.1.4.1.912.1.7.0

scalarIndex

DESCRIPTION: The index of the variable currently selected. This field is only significant for devices that contain data that can be represented as a scalar (see *Scalar and Boolean Class Variables and Devices* for more information).

TYPE: Integer greater than 0, whose upper bound is determined by **numberOfScalarVariables**; that is, you cannot refer to scalar variable number 4 if the connected device contains only two variables.

ACCESS: Read/Write

OID .1.3.6.1.4.1.912.1.8.0

scalarVarName

DESCRIPTION: The name of the Scalar variable currently selected (e.g., “Temperature”). This is retrieved from the SSD, and cannot be modified.

TYPE: String

ACCESS: Read only

OID .1.3.6.1.4.1.912.1.9.0

scalarData

DESCRIPTION: This field contains the data of the currently selected Scalar variable. The index of the variable being selected is described by the Agent variable **scalarIndex** (described above).

Note that even though this variable has Read/Write access, the SSD has the final word on whether or not the user can modify its variables. Check the variable **scalarRWCcapability** to determine if the variable that is currently being selected has write permission. If it does not, attempts to set **scalarData** will fail.

TYPE: Integer

ACCESS: Read/Write (see above)

OID .1.3.6.1.4.1.912.1.10.0

unitOfMeasure

DESCRIPTION: Describes the unit of measure being used to represent the data retrieved from the currently selected variable.

Specifically, for temperature sensors, the unit of measure can be set to “F” or “C”, causing **scalarData** to represent temperature in Fahrenheit or Celsius, respectively.

If the user enters an invalid value, the default unit of measure will be used, but this field will still display the invalid unit of measurement as entered.

TYPE: String

ACCESS: Read/Write

OID .1.3.6.1.4.1.912.1.11.0

Non-volatile

scalarRWCcapability

DESCRIPTION: Describes the read/write permission of the variable being selected. Possible values are:

<i>RO</i>	Read Only
<i>RW</i>	Read/Write

If this field is *RO*, the Sensorsoft device will not allow write operations on the currently selected variable. Attempts to change **scalarData** or **booleanData** will fail.

TYPE: String (see above)

ACCESS: Read-only

OID .1.3.6.1.4.1.912.1.12.0

scalarCriticalHighThreshold

DESCRIPTION: This is where the user can define the maximum allowable value of **scalarData** before critical traps are issued. If the value in **scalarData** exceeds this threshold, **status** will change to *Critical High* and traps will be issued (provided **scalarCriticalTrapInterval** is greater than 0).

See also *Traps* and *Description of Traps Issued by the Agent*.

TYPE: Integer

ACCESS: Read/Write

OID .1.3.6.1.4.1.912.1.13.0

Non-volatile

scalarCriticalLowThreshold

DESCRIPTION: The minimum allowable value of **scalarData** before critical traps are issued. If the value of **scalarData** falls below this threshold, **Status** will change to *Critical Low* and traps will be issued (provided **scalarCriticalTrapInterval** is greater than 0).

See also *Traps* and *Description of Traps Issued by the Agent*.

TYPE: Integer

ACCESS: Read/Write

OID .1.3.6.1.4.1.912.1.14.0

Non-volatile

scalarWarningHighThreshold

DESCRIPTION: The maximum allowable value of **scalarData** before warning traps are issued. If the value of **scalarData** exceeds this threshold, **Status** will change to *Warning High* and traps will be issued (provided **scalarWarningTrapInterval** is greater than 0).

See also *Traps* and *Description of Traps Issued by the Agent*.

TYPE: Integer

ACCESS: Read/Write

OID .1.3.6.1.4.1.912.1.15.0

Non-volatile

scalarWarningLowThreshold

DESCRIPTION: The minimum allowable value of **scalarData** before warning traps are issued. If the value of **scalarData** falls below this threshold, **Status** will change to *Warning Low* and traps will be issued (provided **scalarWarningTrapInterval** is greater than 0).

See also *Traps* and *Description of Traps Issued by the Agent*.

TYPE: A number
ACCESS: Read/Write
OID .1.3.6.1.4.1.912.1.16.0
Non-volatile

scalarCriticalTrapInterval

DESCRIPTION: The rate at which critical traps are sent, represented in minutes between traps (hence, the maximum trap rate is 1 per minute). In order to receive traps this must be 1 or greater. A value of zero indicates that critical traps are disabled and will not be sent.

TYPE: Integer, 0 or greater.
ACCESS: Read/Write
OID .1.3.6.1.4.1.912.1.17.0
Non-volatile

scalarWarningTrapInterval

DESCRIPTION: The rate at which warning traps are sent, represented in minutes between traps (hence, the maximum trap rate is 1 per minute). A value of zero indicates that warning traps are disabled and will not be sent.

See *Description of Traps Issued by the Agent*

TYPE: Integer, 0 or greater.
OID .1.3.6.1.4.1.912.1.18.0
Non-volatile

numberOfBooleanVariables

DESCRIPTION: The number of Boolean variables in the currently connected Sensorsoft Device. The specific variable being selected is represented by **booleanIndex**.

See also *Scalar and Boolean Class Variables and Devices*.

TYPE: Integer, range 0-6.
ACCESS: Read only
OID .1.3.6.1.4.1.912.1.19.0

booleanIndex

DESCRIPTION: The index of the variable currently selected. This field is only significant for devices that contain data that can be represented as a Boolean (see *Scalar and Boolean Class Variables and Devices* for more information).

TYPE: Integer greater than 0, whose upper bound is determined by **numberOfBooleanVariables**; that is, you cannot refer to variable number 4 if the connected device contains only two Boolean variables.

ACCESS: Read/Write
OID .1.3.6.1.4.1.912.1.20.0

booleanVarName

DESCRIPTION: The name of the variable currently being selected (e.g., "POWER_BIT").

TYPE: String

ACCESS: Read only

OID .1.3.6.1.4.1.912.1.21.0

booleanData

DESCRIPTION: This field contains the data of the currently selected variable, provided that variable is Boolean (see *Scalar and Boolean Class Variables and Devices*). The variable selected is described by the variable **booleanIndex**. The data takes the form of a string representing the state, such as ON or OFF, or PWR OK or PWR FAIL.

Note that even though this variable has read/write access, the SSD has the final word on whether or not the user can modify its variables. Check **booleanRWCapability** to determine if the variable that is currently being selected has write permission. If it does not, attempts to set **booleanData** will fail.

For Boolean variables with write access, see the variables **booleanOneStateString** and **booleanZeroStateString** to determine the valid values for **booleanData**.

TYPE: String

ACCESS: Read/Write

OID .1.3.6.1.4.1.912.1.22.0

booleanOneStateString

DESCRIPTION: A string describing one of the states of the variable currently being monitored. Its counterpart is stored in **booleanZeroStateString**. The values in these variables can be used to set the value of **booleanData**, provided the current variable has write access.

TYPE: String

ACCESS: Read only

OID .1.3.6.1.4.1.912.1.23.0

booleanZeroStateString

DESCRIPTION: A string describing one of the states of the variable currently being monitored, provided that variable is Boolean. Its counterpart is stored in **booleanOneStateString**. The values in these variables can be used to set the value of **booleanData**, provided the current variable has write access.

TYPE: String

ACCESS: Read only

OID .1.3.6.1.4.1.912.1.24.0

booleanRWCapability

DESCRIPTION: Describes the read/write permission of the variable being selected, provided that variable is Boolean. Possible values are as follows.

RO Read Only

RW Read/Write

If this field is *RO*, the Sensorsoft device will not allow write operations on the currently selected variable. Attempts to change the field **booleanData** will fail.

TYPE: String (see above)

ACCESS: Read-only

OID .1.3.6.1.4.1.912.1.25.0

booleanCriticalState

DESCRIPTION: Describes which of the Boolean states will cause Boolean critical traps to be issued when **booleanData** takes on that value. To obtain the possible states, query **booleanZeroStateString** and **booleanOneStateString**.

TYPE: String

ACCESS: Read/Write

OID .1.3.6.1.4.1.912.1.26.0

Non-volatile

booleanCriticalTrapInterval

DESCRIPTION: The rate at which critical traps are sent when **status** is *Critical State* (that is, when **booleanData** is the same as **booleanCriticalState**). This is represented in minutes between traps (hence, the maximum trap rate is 1 per minute). A value of zero indicates that Boolean critical traps are disabled and will not be sent.

TYPE: Integer, 0 or greater.

ACCESS: Read/Write

OID .1.3.6.1.4.1.912.1.27.0

Non-volatile

Description of Traps Issued by the Agent

The following is a list of traps that are issued by the agent. The “Num” column contains the trap specific identification number that can be used to determine the nature of the trap as sent by the Agent. For more information on traps, see *Traps*.

Num	Name	Description
1	zeroCritical	Issued when booleanData is in the zero state (specified by booleanZeroStateString) and the user has specified that as a critical state. Generated at the rate specified in booleanCriticalTrapInterval .
2	oneCritical	Issued when booleanData is in the one state (specified by booleanOneStateString) and the user has specified that as a critical state. Generated at the rate specified in booleanCriticalTrapInterval .
3	criticalHigh	Issued when scalarData exceeds the user-specified high critical threshold. Generated at the rate specified in scalarCriticalTrapInterval .
4	criticalLow	Issued when scalarData falls below the user-specified low critical threshold. Generated at the rate specified in scalarCriticalTrapInterval .
5	warningHigh	Issued when scalarData exceeds the user-specified warning high threshold but does not violate the critical thresholds. (If both a warning state and critical state are occurring, only critical traps will be generated). Generated at the rate specified in scalarWarningTrapInterval .
6	warningLow	Issued when the scalarData falls below the user-specified warning low threshold but does not violate the critical thresholds. (If both a warning state and critical state are occurring, only critical traps will be generated). Generated at the rate specified in scalarWarningTrapInterval .
7	sensorRemoved	Issued when the Agent can no longer communicate with the SSD. This trap is issued only once.
8	newSensor	Issued when the Agent detects the presence of a new SSD. This trap is issued only once, when the new SSD is first connected.

Troubleshooting Guide

The troubleshooting guide is designed to help you in the event that you should have trouble with your Sensorsoft SNMP Agent. It is composed of the following sections:

Troubleshooting Checklist is a list of things that are easy to verify and easy to remedy. This is the first place you should look for a solution to your problems.

The *Common Problems* section provides a list of common problems and their solutions.

Restarting the Sensorsoft SNMP Agent provides information on determining if the Sensorsoft SNMP Agent has stopped running and how to restart it.

In order to make effective use of this guide (and Sensorsoft technical support, should it be necessary) it is beneficial that you attempt to determine the precise nature of the problem. For example, ping the MSS to make sure it is accessible on the network, and verify that your NMS works correctly with other (non-Sensorsoft) SNMP Agents.

Troubleshooting Checklist

If you are having difficulty communicating with the Sensorsoft SNMP Agent, please ensure that all of the following conditions are met. Violating one or more of the following requirements can cause any number of problems.

- The MSS is connected to a powered outlet using the approved AC adapter and the POWER light is illuminated.
- The MSS is connected to an Ethernet hub or concentrator using a known good UTP patch cable and the LINK light is illuminated.
- You can ping the MSS to ensure it is available on the network. If it does not respond to pings, then your first priority should be to establish a valid network connection.
- You are giving the MSS at least 2-3 minutes to boot and run the Sensorsoft SNMP Agent software.
- A Sensorsoft device is connected to the MSS's serial port. The SERIAL light should flash red every 3-4 seconds.
- You are using a timeout value of at least 6 seconds (6000 milliseconds) in your NMS software.
- You are using the community name *public*; the Sensorsoft Agent does not support other community names.
- Make sure you are using the variables appropriate to the class of SSD attached (see *Scalar and Boolean Variables and Devices* for more information).
- If your NMS software is running under Microsoft Windows, be sure that the appropriate SNMP Services are running.

Common Problems

Many of the problems covered in this section can also be caused by violating one of the items presented in the previous section, *Troubleshooting Checklist*. The answers given here omit some solutions in an effort to avoid redundant information. If this section does not solve your problem, please consult the other sections of this troubleshooting guide.

GET and SET requests are failing repeatedly.

See *Troubleshooting Checklist*, which provides a list of many common reasons for GET and SET requests to fail.

Some GET requests succeed, but other's fail intermittently.

Make sure your NMS software timeout value is set to at least 6 seconds (equivalent to 6000 milliseconds).

I get *sensorRemoved* and *newSensor* traps, however critical or warning traps are not being generated.

First, verify that the SNMP agent is indeed in an alert state by checking the value of **status**. If it does not indicate a critical or warning state, make sure your thresholds are set correctly (or, if a Boolean sensor is being used, make sure **booleanCriticalState** is set to one of the strings in **booleanZeroStateString** or **booleanOneStateString**).

Secondly, there are three variables for controlling the trap interval. Make sure you have the appropriate variable set to a value of one or greater.

I am not receiving any traps from the Sensorsoft SNMP Agent.

An easy way to test for traps from the Sensorsoft SNMP Agent is to unplug the SSD or reconnect it to the MSS serial port.

If you have other SNMP agents, verify that you can receive traps from them. If you are not receiving traps from any of your agents, the solution may be outside the scope of this document.

Verify the following:

- Is the **trapDesintation** variable on the Sensorsoft SNMP Agent set to the an IP address that is accessible to the Sensorsoft SNMP Agent?
- Is your NMS software properly configured to receive traps (consult the NMS documentation).
- If you're NMS software is running on Microsoft Windows, check that the *SNMP Trap Service* is running.

In Microsoft Windows 2000 you may encounter a problem whereby traps might not be received correctly. This bug is fixed in Windows 2000 Service Pack 1. If you find that your NMS software is not receiving traps from any source, upgrade to the above Microsoft Service Pack.

The NMS software is saying that variables are not available, null, or that they contain no data.

Once the MSS has booted, it must run the Sensorsoft SNMP Agent software, during which GET requests may succeed but the data cannot be retrieved. The entire process of booting the MSS and running the Sensorsoft SNMP Agent software takes approximately 2-3 minutes.

I changed the value of *unitOfMeasure*, but *scalarData* did not change or does not appear to be using the new units.

The unit of measure you entered was not recognized by Sensorsoft SNMP Agent. No restriction is placed on the value of **unitOfMeasure**; the Agent will happily update **unitOfMeasure** regardless of whether it is valid or not. However, if it does not know how to represent the data in terms of **unitOfMeasure**, it will use the default unit of measurement.

**I have *booleanIndex* set to 1, but *booleanData* is reading “NA”.
I have *scalarIndex* set to 1, but *scalarData* is reading 0, which I know is incorrect.**

Make sure you have the appropriate class of SSD attached (Boolean or Scalar). The Agent will report 1 as the Scalar/Boolean index even if the number of variables of that type is 0. Also, check **status** to make sure it is not *Offline*. If it is, connect an SSD to the MSS’s serial port.

**When I try to GET the Location MIB variable, only part of it is returned.
When I try to SET the Location MIB variable, my NMS software only displays part of the string entered.**

The Location MIB variable has a size limit of 79 characters. If you attempt to enter a string that is longer, the text will be truncated by the Agent.

SET commands are not working: the Agent retains the old value.

For numerical variables (and **booleanData**), entering an invalid value will not have an effect. Ensure the value you are entering is valid.

***trapDestination* is showing 13.0.0.0 or 255.255.255.255 no matter what I change it to.
When I enter a *trapDestination*, the GET succeeds and the value is changed, but not to what I specified.**

Make sure you are entering a valid IP address. Because of the way IP addresses are read and stored by the agent, invalid addresses may be misinterpreted. For example, 53.4.4 will be interpreted as 13.0.0.0.

Most SET requests work, but when I try to set *scalarData* or *booleanData*, it fails.

SET request on **scalarData** or **booleanData** will fail if you do not have write access to the variable. Note that the write access of the variable **scalarData** or **booleanData** as reported by the Sensorsoft SNMP Agent does *not* indicate the write access of the SSD’s variable. Query the value of **scalarRWCcapability** or **booleanRWCcapability**, appropriately, to make sure you have

write access. Also make sure **scalarIndex** or **booleanIndex** is set to the index of the variable you are trying to modify.

In the case of Boolean devices like the Sensorsoft Relay, a SET request on **booleanData** will fail if the given value is not one of **booleanZeroStateString** or **booleanOneStateString**. Ensure that you are entering the string *exactly* as it appears under one of those variables.

I want to control a Sensorsoft Relay using the Sensorsoft SNMP Agent

To control a Sensorsoft Relay, SET **booleanData** to ON or OFF.

What settings are valid for *booleanData*?

The valid strings for **booleanData** appear in the **booleanZeroStateString** and **booleanOneStateString**.

CAUTION NOTICE

The Lantronix MSS100-01-SSA with Version 3.6/3 firmware was specially setup to run the Sensorsoft SNMP Agent software.

Do not upgrade the Lantronix firmware or change any Lantronix settings on this unit, with the following exceptions:

IP ADDRESS
SUBNET MASK
NAMESERVER
DOMAIN
GATEWAY
PRIVPASS

Changing any other settings or upgrading the firmware will cause the Sensorsoft SNMP Agent to stop running and will need to be sent back to the factory.

If you are having extreme difficulty setting your IP address and need to reset the unit to factory default, please refer to the Sensorsoft SNMP Agent User Manual for instructions on how to restart the Sensorsoft SNMP Agent software.

Restarting the Sensorsoft SNMP Agent software

This section provides details of how to determine if the Sensorsoft SNMP Agent has stopped running and how to restart it.

This procedure only needs to be executed if the results of the *Troubleshooting Checklist and Common Problems* did not contain information on how to solve the problem:

Determining if the Sensorsoft SNMP Agent has stopped running

Telnet into the MSS100, logging in with any user name. At the prompt, type the following command:

```
Local_1>> show ports
```

Verify that you have the following settings:

```
Baud Rate: 1200
Dedicated SDK: /oem/snmp.c (field below Port name: )
```

If both of the above conditions do not hold, the MSS's settings have been modified or reset to factory default. In this case execute the following procedure.

Procedure for restarting the Sensorsoft SNMP Agent

The following procedure can be used to restart the Sensorsoft SNMP Agent in the event that settings critical to its operation are modified. To determine if it is necessary to do this, see the previous procedure.

- 1) Telnet into the MSS, logging in with any user name.
- 2) Log in as a privileged user by entering the following:

```
Local_1>> SET PRIV
Password> system (not echoed)
```

- 3) Enter the following commands:

```
Local_1>> CHANGE AUTOBAUD DISABLED
Local_1>> CHANGE FLOW CONTROL NONE
Local_1>> CHANGE MODEM CONTROL DISABLED
Local_1>> CHANGE INACTIVE LOGOUT ENABLED
Local_1>> CHANGE INACTIVE TIMER 30s (specifies 30 seconds)
Local_1>> CHANGE DSRLOGOUT DISABLED
Local_1>> CHANGE SIGNAL CHECK DISABLED
Local_1>> CHANGE DTRWAIT DISABLED
Local_1>> CHANGE ACCESS DYNAMIC
Local_1>> CHANGE PARITY NONE
Local_1>> CHANGE STOPBITS 1
Local_1>> CHANGE CHARSIZE 8
```

```
Local_1>> CHANGE SPEED 1200
Local_1>> CC -AUTO /OEM/SNMP.C
```

- 4) Reboot the MSS by entering the following command:

```
Local_1>> INIT DELAY 0
```

Your telnet session will now be disconnected and the MSS will reboot. Allow 2-3 minutes for the Sensorsoft SNMP Agent software to start up.

Stopping the Sensorsoft SNMP Agent software

If you wish to use the MSS100 with Sensorsoft SCOM or Remote Watchman Client software you must stop the Sensorsoft Agent from running. Follow the procedure below to do this:

- 1) Telnet into the MSS, logging in with any user name.
- 2) Log in as a privileged user by entering the following:

```
Local_1>> SET PRIV
Password> system          (not echoed)
```

- 3) Enter the following command:

```
Local_1>> CC -NOAUTO
```

- 4) Reboot the MSS by entering the following command:

```
Local_1>> INIT DELAY 0
```

Your telnet session will now be disconnected and the MSS will reboot. After a minute or two the unit will come up and the Sensorsoft Agent will not be running. You can now connect to the MSS100 using Sensorsoft SCOM or Remote Watchman Client software (use TCP port 3001).

To restart the Sensorsoft Agent (to enable SNMP access) execute the above procedure *Restarting the Sensorsoft SNMP Agent software*.

Getting Help

Technical Support

If in the unlikely event you should have problems installing or using the Sensorsoft SNMP Agent, and the previous sections have failed to provide a solution, we offer technical support to help you overcome your difficulties.

No-charge installation and configuration support is provided for 90 days, after which there is a per-incident fee. Sensorsoft *does not* provide free support to those wishing to write their own software; this is available on a charge-per-incident basis only.

Before contacting support:

Go through the *Troubleshooting Guide* in the previous sections of this manual. Even if a direct answer to your question is not found there, it will be helpful for the support technician if you are able to provide information obtained from the diagnostic and troubleshooting process. Please ensure the problem is directly related to the Sensorsoft SNMP Agent software. We do not provide support for specific network management software.

World Wide Web: <http://www.sensorsoft.com/company.html>
E-mail: support@sensorsoft.com

30 Day No Risk Guarantee

If for any reason you want to return a Sensorsoft product for a refund, you can do so within 30 days (calendar days) of your purchase. The refund does not include shipping or brokerage fees you may have incurred or paid.

Returns

If returning a product or item, please keep in mind the follow guidelines:

- Contact Sensorsoft for an RMA number (Return Material Authorization).
- Provide a detailed explanation or reason for returning the product.
- Return shipments that bear no RMA number on the outside of the package will not be accepted.