

SECTION I

## Introduction

This introduction is divided into two sections. The first section, "Introduction to the SNMP Reference Guide," explains the *SNMP Reference Guide* design. All essential Simple Network Management Protocol (SNMP) terms are defined in this section. Some of the vocabulary may seem complex and unfamiliar to system administrators who are using SNMP for the first time. SNMP experts can skim this section, and beginners can read the section more carefully.

The second section, "Introduction to the Server Administrator SNMP Subagent," is a more technical introduction to the management information base (MIB) that underlies Server Administrator services.

## Audience

This guide is intended for system administrators, network administrators, and for anyone who wants to write SNMP MIB applications to monitor systems.

## General Content

This reference guide provides a formatted version of the Server Administrator MIB (filename `10892.mib`) that was released with Dell OpenManage™ Server Administrator 1.1.x or later. Sections in this guide follow MIB groups and provide explanations and definitions for the terms used to define MIB objects.

# Introduction to the SNMP Reference Guide

Content in this reference guide is organized as follows:

**Table 1-1. Content of the Sections in This Guide**

Section	Topics	MIB Group Number
1	Introduction to SNMP basics and to the MIB that supports Server Administrator services	NA
2	MIB Group — defines major and minor version numbers of the MIB	1
3	Systems Management Software Group — defines the supported systems management standards	100
4	System State Group — defines status, state, and redundancy for a system and its components	200
5	Chassis Information Group — defines chassis types, events, and indicators	300

**Table 1-1. Content of the Sections in This Guide (continued)**

<b>Section</b>	<b>Topics</b>	<b>MIB Group Number</b>
<b>6</b>	Operating System Group — defines variables for name, version, service pack, and other information about a system's operating system	400
<b>7</b>	System Resource Group — defines variables for input/output ports, memory, interrupts, and direct memory access	500
<b>8</b>	Power Group — defines variables for power units, power supplies, and their current and voltage probes	600
<b>9</b>	Thermal Group — defines variables for temperature probes and cooling devices	700
<b>10</b>	User Security Group — defines variables for creating and modifying user accounts	800
<b>11</b>	Remote Flash BIOS Group — defines variables for updating the system's BIOS remotely	900
<b>12</b>	Port Group — defines variables for major port types such as keyboard, monitor, small computer system interface (SCSI), Universal Serial Bus (USB), and parallel and serial ports	1000
<b>13</b>	Device Group — defines variables for pointing, keyboard, processor, cache, memory, and personal computer interface devices	1100
<b>14</b>	Slot Group — defines variables for the system's slots	1200
<b>15</b>	Memory Group — defines variables for the system's physical memory	1300
<b>16</b>	BIOS Setup Control Group — defines variables for BIOS functions such as boot sequence, speakers, Wake on the local area network (LAN), diskettes, ports, and network interface controllers (NIC)	1400
<b>17</b>	Local Response Agent Group — defines variables for global settings and actions. These variables allow users to predetermine how the system responds to a particular type of event	1500
<b>18</b>	Cost of Ownership Group — defines variables for tracking data on the system's service contract, lease, repair records, trouble tickets, and so on	1600
<b>19</b>	Remote Access Group — provides information about remote access hardware that may be present in your system and defines variables for administrative users, SNMP trap destinations, modem configuration for dial-up networking, dial-in configuration, and dial-out destinations	1700
<b>20</b>	Cluster Group — defines variables for systems that operate as a cluster	1800
<b>21</b>	Traps — defines the types of alerts that can be sent to report the status of critical components	5000

**Table 1-1. Content of the Sections in This Guide (continued)**

Section	Topics	MIB Group Number
<b>Glossary</b>	Defines acronyms, abbreviations, and technical terms used in this reference guide	NA
<b>Appendix A</b>	Defines standard data types used in this reference guide	NA
<b>Appendix B</b>	Provides a sample SNMP output	NA

## How This Guide Defines Technical Terms

The following table provides information about where to find definitions for technical terms in this reference guide.

**Table 1-2. Where to Find Definitions for Technical Terms**

Type of Definition	See
Basic SNMP vocabulary.	Introduction
MIB-group-specific variable values. MIB-group-specific MIB variables contain links to the tables that define these values in the last section of the section in which these variables are used.	Sections 3, 5, 7, 8, 9, and 11 through 18.
Systems management terms, acronyms, and commonly managed components referred to in this reference guide.	Glossary
Server Administrator-standard data types that specify variable values in this reference guide.	Appendix A, "Standard Data Type Definitions."


## SNMP Basic Terminology

It is important to have a good understanding of the key technical terms used in this guide. This guide provides definitions for all essential terms used in describing the Server Administrator MIB.

The Glossary contains definitions for all essential terms and acronyms.

### Managed Object

A managed object is any item in a computer system that can be singled out for discovery, monitoring, or user intervention and correction.

 **NOTE:** Not all managed objects described in this guide are supported by all systems.

## **MIB**

A MIB acts as a structured road map for managed objects. As an Application Programming Interface (API), a MIB allows systems management tools to retrieve data maintained by an agent. The server administrator MIB is divided into several major groups of managed objects.

## **Variable**

A variable is a component of a managed object. A temperature probe, for example, has a variable to describe its capabilities, its health or status, and certain indexes that you can use to locate specific temperature probes. One index for the probe would be the probe's chassis number. Some systems may have multiple chassis—one chassis for the central processing unit and another chassis for storage. A chassis within a system can also have more than one temperature probe. Variables for a temperature probe include its capabilities, status, chassis index, and index.

## **One-Based Index**

When an index is one-based, counting starts at 1. Zero-based indexing begins counting at 0, followed by 1, 2 and so on.

## **Zero-Based Index**

When an index is zero-based, counting starts at 0. Zero-based indexing counts the first instance as 0, the second index as 1, and so on.

## **Fields**

Managed object variables contain fields. In this reference guide, managed object variables have the following fields defined:

**Name** is the exact string by which the variable is known in the MIB. MIB variables are named according to the following conventions:

- Variable names start with a lowercase letter.
- Spaces are not allowed between words in the variable name.
- Acronyms are in uppercase letters, except when an acronym is the first word in the variable name.
- With the exception of the first letter of the variable name and acronyms, all other words in the variable name start with capital letters.

The following variable names illustrate these conventions:

```
temperatureProbeLowerCriticalThreshold  
coolingUnitIndex  
pCIDeviceSpeed
```

**Object Identifier (OID)** is the unique number assigned to an object defined in a MIB. An OID is written as a sequence of subidentifiers in decimal notation. Each OID in this reference guide has a prefix that identifies the managed objects as belonging to Dell: 1.3.6.1.4.1.674. The additional numbers identify the MIB group and subgroup as well as the table entry number of any variables.

For example, the OID for the temperature probe managed object table is 700.20 and the variable for the location of the temperature probe (temperatureProbeLocationName) has an OID of 700.20.1.8. The full OIDs for these items are 1.3.6.1.4.1.674.10892.1.700.20 for the temperatureProbeTable and 1.3.6.1.4.1.674.10892.1.700.20.1.8 for the temperatureProbeLocation. For more information about the structure of OIDs, see "SNMP MIB OIDs."

**Description** is a brief explanation of what a particular managed object does.

**Syntax** defines the data type in which the values of the variable must be expressed. Most variables in this reference guide use standard data types such as string or boolean. All data types that are unique to server administrator variables are defined at the end of the section in which they occur. Standard data types are defined in "Standard Data Type Definitions."

**Access** specifies whether persons with administrative privileges can read but not modify the value of a variable (read only) or can both read and modify the value of a variable (read-write).

## Frequently Used Terms in Variable Names

The following terms are frequently used in the name of a MIB variable:

**Capability** refers to the actions an object can perform, or to actions that can be taken by the object. Hot-pluggable is an example of a capability. If a card is hot-pluggable, it can be replaced while a system is running. Capability settings refer to the capabilities of the object that the user can select from and activate if desired. Capability settings allow users of the server administrator to predetermine how an object will behave under specific conditions.

**Settings** are the conditions of a manageable object that determine what happens when a certain value is detected in a component. For example, a user can set the upper critical threshold of a temperature probe to 75 degrees Celsius. If the probe reaches that temperature, the setting causes an alert to be sent to the management console. Some settings, when reached, can trigger a system shutdown or other response to prevent damage to the system.

**State** refers to the condition of an object that has more than one condition. For example, an object may be in a “not ready” or in an “enabled” state.

**Status** refers to the health of an object or how the object is functioning. For example, the status of a temperature probe that is measuring acceptable temperatures would be reported as normal. When the probe begins reading temperatures that exceed limits set by the user, it reports a critical status.

## **Tables**

This reference guide contains two types of tables: tables that are used to organize and define variable values and tables that define MIB objects. Readers must understand the differences between these two types of tables.

### **SNMP Tables**

Most of the MIB objects defined in this reference guide are organized into SNMP tables. SNMP tables organize data into two-dimensional structural arrays. In SNMP, objects that have a relationship to other objects are called columnar objects. Columnar objects are the type of object used to form lists and tables. When a MIB group is divided into one or more discrete tables, the word "table" has a technical meaning. An example is the section of this reference guide entitled Universal Unique Identifier (UUID). The UUID object has a type and a value that uniquely identify an object such as a chassis. The table defines all of the variables that comprise the managed object UUID.

The following table is an example of an SNMP table. The table contains variables that must occur in a definite sequence. In the example table the defined variables are UUID Chassis Index, UUID Index, UUID Type, and UUID Value.

## Example SNMP Table

### *UUID Table*

These objects comprise the Server Administrator definitions for the UUID.

<b>Name</b>	uUUIDTable
<b>Object ID</b>	1.3.6.1.4.1.674.10892.1.300.20
<b>Description</b>	Defines the UUID table.
<b>Syntax</b>	SEQUENCE OF UUIDTableEntry
<b>Access</b>	Not accessible

### *UUID Table Entry*

<b>Name</b>	uUUIDTableEntry
<b>Object ID</b>	1.3.6.1.4.1.674.10892.1.300.20.1
<b>Description</b>	Defines the UUID table entry.
<b>Syntax</b>	UUIDTableEntry
<b>Access</b>	Not accessible
<b>Index</b>	uUUIDIndex, uUUIDchassisIndex

### *UUID Chassis Index*

<b>Name</b>	uUUIDchassisIndex
<b>Object ID</b>	1.3.6.1.4.1.674.10892.1.300.20.1.1
<b>Description</b>	Defines the index (ones-based) of this chassis.
<b>Syntax</b>	DellObjectRange
<b>Access</b>	Read-only

### *UUID Index*

<b>Name</b>	uUUIDIndex
<b>Object ID</b>	1.3.6.1.4.1.674.10892.1.300.20.1.2
<b>Description</b>	Defines the index of the UUID in a specified chassis.
<b>Syntax</b>	DellObjectRange
<b>Access</b>	Read-only


### **UUID Type**

<b>Name</b>	uUUIDType
<b>Object ID</b>	1.3.6.1.4.1.674.10892.1.300.20.1.3
<b>Description</b>	Defines the type of the UUID for this chassis.
<b>Syntax</b>	DellUUIDType
<b>Access</b>	Read-only

### **UUID Value**

<b>Name</b>	uUUIDValue
<b>Object ID</b>	1.3.6.1.4.1.674.10892.1.300.20.1.4
<b>Description</b>	Defines the value of the UUID for this chassis.
<b>Syntax</b>	OCTET STRING (SIZE[16])
<b>Access</b>	Read-only

## **Reference Guide Content Tables**

 **NOTE:** Variable values are defined for any variable that is Server Administrator-specific. Industry-standard variable definitions are documented in "Standard Data Type Definitions."

Some of the tables in this guide have no technical significance in SNMP. These tables are designed to show information in a readable form. The following table, for example, defines the Server Administrator-specific variable, DellFanControlCapabilities. The table provides the name of the variable, its data type, the values that are valid for the variable, and the meaning of each value.

**Table 1-3. Example Variable Type Definition Table**

---

<b>Variable Name:</b> DellFanControlCapabilities	
<b>Data Type:</b> Integer	
<b>Possible Data Values</b>	<b>Meaning of Data Value</b>
unknown(1)	The fan's capabilities are unknown.
lowSpeedCapable(2)	The fan can be set to low speed.
highSpeedCapable(4)	The fan can be set to high speed.
lowOrHighSpeedCapable(6)	The fan can be set to low or high speed.

---

This type of table is used throughout the reference guide to list and define variable values. Tables that explain Server Administrator-specific variable values are located in the final section of sections that define Server Administrator-specific variables. In the preceding example, the variable name is `DellFanControlCapabilities`. This variable must be expressed as an integer and has four possible values: `unknown`, `lowSpeedCapable`, `highSpeedCapable`, and `lowOrHighSpeed Capable`.

## Section Organization

Sections in this reference guide are based on the Server Administrator MIB, so the complexity of each section depends on the complexity of each MIB group. The first section provides a high-level introduction to the MIB group. If the group is defined by one or more tables, the second section lists these tables. The third section documents the variables that comprise the group, and if applicable, the variables that comprise the tables. The fourth section contains definitions for any Server Administrator-specific variables that are used in the section. The following example shows the typical content of these four sections.

### 1 BIOS Setup Control Group

This section explains the purpose of the MIB group and summarizes the major features of the component groups.

### 2 BIOS Group Tables

If there is more than one SNMP table for a group, this section lists all of the tables. For this BIOS group example, there are eight tables listed. Double-clicking any table on the list takes you to that table.

- BIOS Setup Control Table
- SCSI Control Table
- Parallel Port Control Table
- Serial Port Control Table
- USB Control Table
- IDE Control Table
- Diskette Control Table
- Network Interface Control Table

### 3 Variables that make up each table in the group

This section documents the variables for the eight tables that comprise the BIOS group.

#### 4 BIOS Variable Values

This section explains any Server Administrator-specific variables and data types that are used in this section. In the BIOS group example, there are 17 unique, Server Administrator-specific variable meanings. Information on each Server Administrator-specific variable is presented in a formatted table.

### Other Documents You May Need

In addition to this *Server Administrator SNMP Reference Guide*, you can find the following guides on your documentation CD:

- The *Server Administrator Messages Reference Guide* lists the messages that you can receive on your systems management console or on your operating system's event viewer. This guide explains the text, severity, and cause of each message that the server administrator issues.
- *Server Administrator CIM Reference Guide* documents the Common Information Model (CIM) provider, an extension of the standard management object format (MOF) file. The Server-Administrator provider documents supported classes of management objects.

## Introduction to the Server Administrator SNMP Subagent

This guide provides formatted information drawn primarily from the MIB file written for the server administrator SNMP subagent. The MIB filename is `10892.mib`.

For each of the variables defined in the MIB, the following fields are specified:

- Variable name
- OID or unique identifying number
- Description
- Data type of the variable (for example: integer, string, octet string)
- Whether the variable is accessible, not accessible, read-only, or read-write
- Index or indexes, if applicable

For each MIB group that has unique variable definitions, tables are included in the last section of the section to explain the meaning of the terms.

SNMP is a systems management standard originally designed for network management. SNMP manages much more than networks. Information Technology (IT) professionals use SNMP for monitoring and managing computer systems and the various components and peripherals supported by their systems.

SNMP standards are defined by the Internet Engineering Task Force (IETF). SNMP version 1 was published in August 1988 and is the most commonly supported version of SNMP. SNMP version 2 was first published in May 1993, but has not gained widespread market acceptance. SNMP version 3 was recently completed and has addressed security issues that exist in version 1.

All SNMP systems consist of one or more managed nodes that provide data through an SNMP agent to a management station. The management station provides a user interface to view data from the managed nodes. The management station and managed nodes communicate over a network (typically through User Datagram Protocol/Internet Protocol [UDP/IP]).

The management station and a managed node communicate by means of a common data schema. SNMP MIB files define the structure, type, and values of the SNMP data. While MIBs can be standardized or enterprise specific, most operating systems supply SNMP agents for the standard MIB-I and MIB-II schemas. MIB-I defines a base set of standard management information for systems implementing the Internet Protocol (IP) suite. MIB-II defines characteristics of the system, characteristics of network interfaces, and characteristics of components of the IP on the system. In addition to the standard MIBs, many hardware vendors have defined MIBs that provide management data specific to their systems and peripheral devices.

Monitored data can be retrieved through SNMP using the Get command. Typically, this command requires the host name or IP address of the target machine as well as the OID of the data to retrieve. Exact details are dependent on the operating system and the development tools being used to create the management application. The Get command has a variant known as GetNext.

## SNMP MIB OIDs

Each data class within an MIB is defined by an OID. OIDs are unique across all MIBs. An OID consists of a series of digits separated by periods. The OID functions in a similar fashion to a phone number. The phone number 011-512-471-0000 uniquely identifies a single phone. The phone number can be broken down into a number of components to uniquely identify a phone. The first component, 011, is the country code for the United States. The second component, 512, identifies the area code for central Texas. The third component, 471, is the phone exchange for a large state university in the city of Austin. The final component, 0000, is the main switchboard.

There are two main differences between the phone number example and an actual OID. The first difference is that there are many more components in an OID, up to 128. The combination of these components is called an OID prefix. The second difference is that OIDs support the concept of indexes or keys. The OID prefix specifies the data class but does not specify an instance of the data within the class. Indexes can be used to identify the instances of a data class. These indexes are referred to as the OID suffix.

The assignment of values for each OID prefix component can be illustrated by using a tree structure. The following is an example of an OID assignment:

```
ROOT
  CCITT (0)
  ISO (1)
    ORG (3)
      DOD (6)
        INTERNET (1)
          MGMT (2)
            MIB (1)
              EXPERIMENTAL (3)
                PRIVATE (4)
                  ENTERPRISES (1)
                    DELL (674)
                      SNMPv2 (6)
```

In the above example, the OID prefix for the Dell™ enterprise would be 1.3.6.1.4.1.674.

The numbers in boldface type show the categories and numbers that apply to server administrator. All server administrator-defined OIDs consist of 1.3.6.1.4.1.674 followed by additional component values.

## **SNMP Security**

SNMP version 1 has a very limited security mechanism. SNMP agents support the use of a community string, which is configured at each SNMP agent and is passed as a part of all SNMP request messages. There is no verification that the requester is actually a member of the specified community.

Because most system and network management data is not confidential, this limited security is acceptable for Get types of requests. On the other hand, this security is not acceptable for Set types of operations where an SNMP request could power off a system, reconfigure a redundant array of independent disks (RAID) card, and so on. Some vendors have chosen not to support SNMP Set operations for this reason. Server Administrator is able to support SNMP Set operations because its SNMP agents implement a hash/digest mechanism to prevent unauthorized SNMP Set operations. One limitation of this practice is that only server administrator-developed SNMP management applications have the capability to support the hash/digest mechanism.

## **Initiating Management Actions**

Management actions can be performed using the SNMP Set command. These actions can consist of configuring a phone number for the system's owner, rebooting a system, or changing the asset tag of the system. See the previous section, "SNMP Security," for limitations on Set operations.

## **SNMP Traps**

SNMP is frequently used to monitor systems for fault conditions such as temperature violations, hard drive failures, and so on. Management applications can monitor for these conditions by polling the appropriate OIDs with the Get command and analyzing the returned data. This method has its drawbacks. If it is done frequently, significant amounts of network bandwidth can be consumed. If it is done infrequently, the response to the fault condition may not occur in a timely fashion. SNMP traps avoid these limitations of the polling method.

An SNMP trap is an asynchronous event indicating that something significant has occurred. This is analogous to a pager receiving an important message, except that the SNMP trap frequently contains all the information needed to diagnose a fault.

Two drawbacks to SNMP traps are that they are sent using UDP, which is not a guaranteed delivery mechanism, and that they are not acknowledged by the receiver.

An SNMP trap message contains the trap's enterprise OID, the agent IP address, a generic trap ID, the specific trap ID, a time stamp, and zero or more variable bindings (varbinds). The combination of an enterprise OID and a specific trap ID uniquely identifies each Server Administrator-defined trap. A varbind consists of an OID and its value and provides additional information about the trap.

In order for a management station to receive SNMP traps from a managed node, the node must be configured to send traps to the management station. Trap destination configuration is dependent on the operating system. When this configuration is done, a management application on the management station can wait for traps and act on them when received.

For a list of traps supported by the server administrator SNMP subagent, see "Traps."

