

SNMP Troubleshooting TechNote



Document ID: 115927

Contributed by Bill Moody and Curtis Carrington, Cisco TAC Engineers.

Mar 12, 2013

Contents

Introduction

Prerequisites

- Requirements

- Components Used

- Conventions

Managers and Agents

MIBs, Object IDs, and Instances

Applications

MIBs

Tips

RFCs

Related Information

Introduction

This document provides a brief overview of Simple Network Management Protocol (SNMP) and demonstrates how to test SNMP functionality on a Cisco Unified Computing System (UCS) C-Series Rack Server. You can also use it as a reference for basic SNMP troubleshooting.

SNMP is an Internet Engineering Task Force (IETF) industry standard defined by multiple requests for comments (RFCs). SNMP is used in Network Management Systems (NMSs) to monitor network devices. SNMP works at the application layer. The primary purpose of SNMP is to enable NMSs in order to obtain information from managed devices.

Prerequisites

Requirements

This document assumes that the NMS and all network devices have been configured correctly and have worked properly in the past.

Components Used

The information in this document is based on these software and hardware versions:

- SNMP
- Cisco UCS C-Series Rack Servers

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Managers and Agents

The manager is the NMS, and the agent is the software that runs on the managed device. SNMP is used to carry the messages between managers and agents.

MIBs, Object IDs, and Instances

A management information base (MIB) is a collection of objects. There are two types of objects: scalar (single object) and tabular (multiple objects). An object is one or more characteristics or variables of a managed device. These characteristics are identified by an object identifier (OID). An OID is a numerical representation, separated by periods, of the object name and the instance ID. Scalar objects, such as sysDesc, always have an instance of 0. If the object is a column in a table, the instance ID is typically the index of the column. MIBs are structured in a tree format, and object IDs are derived from that tree.

Note: The sample output in this document was captured from a Cisco UCS C-Series Rack Server.

This example looks at the CISCO-UNIFIED-COMPUTING-PROCESSOR-MIB definition for the cucsProcessorUnitEntry.

```
cucsProcessorUnitEntry OBJECT-TYPE
    SYNTAX          CucsProcessorUnitEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION     "Entry for the cucsProcessorUnitTable table."
    INDEX { cucsProcessorUnitInstanceId }
    ::= { cucsProcessorUnitTable 1 }

CucsProcessorUnitEntry ::= SEQUENCE {
    cucsProcessorUnitInstanceId          CucsManagedObjectId,
    cucsProcessorUnitDn                  CucsManagedObjectDn,
    cucsProcessorUnitRn                  SnmpAdminString,
    cucsProcessorUnitArch                CucsProcessorUnitArch,
    cucsProcessorUnitCores                Gauge32,
    cucsProcessorUnitCoresEnabled        Gauge32,
    cucsProcessorUnitId                  Gauge32,
    cucsProcessorUnitModel                SnmpAdminString,
    cucsProcessorUnitOperState            CucsEquipmentOperability,
    cucsProcessorUnitOperability         CucsEquipmentOperability,
    cucsProcessorUnitPerf
CucsEquipmentSensorThresholdStatus,
    cucsProcessorUnitPower                CucsEquipmentPowerState,
    cucsProcessorUnitPresence             CucsEquipmentPresence,
    cucsProcessorUnitRevision             SnmpAdminString,
    cucsProcessorUnitSerial               SnmpAdminString,
    cucsProcessorUnitSocketDesignation    SnmpAdminString,
    cucsProcessorUnitSpeed                INTEGER,
    cucsProcessorUnitStepping             Gauge32,
    cucsProcessorUnitThermal
CucsEquipmentSensorThresholdStatus,
    cucsProcessorUnitThreads              Gauge32,
    cucsProcessorUnitVendor               SnmpAdminString,
    cucsProcessorUnitVoltage
CucsEquipmentSensorThresholdStatus,
    cucsProcessorUnitVisibility           CucsMemoryVisibility,
    cucsProcessorUnitOperQualifierReason  SnmpAdminString
```

cucsProcessorUnitEntry is not accessible, because it is a sequence or parent and is not tied to a specific instance. The objects are listed under the sequence or parent; it may be helpful to think of them as rows in a table. The objects correspond to specific OIDs (numeric object name + instance), which can be solicited individually for information. If you use the **snmpwalk** command on the sequence or parent, you receive replies for all the underlying objects. However, you need to use a translation tool in order to map the object name to the OIDs.

Note:

- Syntax clause defines the format of the integer. For example, counter 32, counter 64, gauge 32, IP address, and so forth.
- Max-access defines read-write, read-create, or no-access.
- Description is self explanatory.
- Index identifies the table ID.

For example, if you use the **snmpwalk** command on the cucsProcessorUnitOperStateObject, you see these results:

```
[root@localhost snmp]# snmpwalk -v2c -c public 14.17.2.121
.1.3.6.1.4.1.9.9.719.1.41.9.1.9
SNMPv2-SMI::enterprises.9.9.719.1.41.9.1.9.1 = INTEGER: 1
SNMPv2-SMI::enterprises.9.9.719.1.41.9.1.9.2 = INTEGER: 1
```

There are two replies to this request, because there are two processors in the server. The replies also reflect the two instances for cucsProcessorUnitOperState, one for each processor. The OID information is pulled for all objects of this type. In this example, the **snmpwalk** command used the path through the MIB tree. In order to translate this, you can load the MIBs locally or search for a tool that translates SNMP MIB objects and OIDs.

This example uses a translation tool in order to translate the OID:

```
.1.3.6.1.4.1.9.9.719.1.41.9.1.9 (path through the tree)
cucsProcessorUnitOperState OBJECT-TYPE
    -- FROM CISCO-UNIFIED-COMPUTING-PROCESSOR-MIB
    -- TEXTUAL CONVENTION CucsEquipmentOperability
    SYNTAX Integer { unknown(0), operable(1), inoperable(2),
degraded(3),poweredOff(4), powerProblem(5), removed(6), voltageProblem(7),
thermalProblem(8), performanceProblem(9), accessibilityProblem(10),
identityUnestablishable(11), biosPostTimeout(12), disabled(13),
fabricConnProblem(51), fabricUnsupportedConn(52), config(81),
equipmentProblem(82), decommissioning(83), chassisLimitExceeded(84),
notSupported(100), discovery(101), discoveryFailed(102), identify(103),
postFailure(104), upgradeProblem(105), peerCommProblem(106),
autoUpgrade(107) }
    MAX-ACCESS read-only
    STATUS Current
    DESCRIPTION "Cisco UCS processor:Unit:operState managed object property"
 ::= { iso(1) org(3) dod(6) internet(1) private(4) enterprises(1) cisco(9)
ciscoMgmt(9) ciscoUnifiedComputingMIB(719) ciscoUnifiedComputingMIBObjects(1)
cucsProcessorObjects(41) cucsProcessorUnitTable(9) cucsProcessorUnitEntry(1) 9 }
```

Above the object type, you can see the path through the MIB tree that was used by the **snmpwalk** command. The response of 1 indicates that the processors are in an operable state.

Applications

In order to use the **snmpwalk** command on a PC, you may need to search for and install an application. You can also search for Windows GUI SNMP applications.

A Mac works natively.

There are packages available for Linux. For Centos or Redhat, Net-SNMP is a command-line application that tests SNMP functionality. In order to install Net-SNMP on Centos or Redhat, use this command:

```
yum install net-snmp-utils
```

Note: If you plan to load the MIBs in order to perform translations locally, create an snmp.conf file and include the desired MIBs.

MIBs

- Cisco UCS C-Series Manager MIB Support List
- Cisco UCS B-Series MIB Support List

Tips

- Ensure the particular MIB is supported in the applicable version of code.
- Verify the IP address of the managed device is reachable.
- Ensure ports are open on firewalls.
- Use Wireshark or TCPDump traces for response and malformed packet issues; filter on IP address and port (UDP port 161 for SNMP, UDP port 162 for recall traps).

RFCs

There are many RFCs associated with SNMP; this is a partial list:

- 1155 Structure and identification of management information for TCP/IP-based internets
- 1156 Management information base for network management of TCP/IP-based internets (MIB I)
- 1157 Simple network management protocol
- 1213 Management information base for network management of TCP/IP-based internets (MIB II)
- 1441 Introduction to version 2 of internet-standard network management framework
- 1452 Coexistence between version 1 and version 2 of the internet-standard network management framework
- 2578 Structure of Management Information Version 2
- 3414 User-based Security Model for version 3 of the Simple Network Management Protocol
- 3584 Coexistence between Version 1, Version 2, and Version 3 of the internet-standard network management protocol

Related Information

- **Technical Support & Documentation – Cisco Systems**
-

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2014 – 2015 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Mar 12, 2013

Document ID: 115927
