



Task 2— Exploring SNMP Capabilities by Using UCD-SNMP

About Using UCD-SNMP

Researching and identifying which functions are available in SNMP are part of building a dial NMS environment. In this case study, UCD-SNMP, an opensource freeware application that allows access to SNMP functions from a command line interface (CLI), is used to explore the capabilities of SNMP.

There are many benefits to using UCD-SNMP.

You can:

- Gain a fundamental understanding of how SNMP functions and protocols work in a dial access environment. This knowledge provides a solid foundation for using automated and GUI-based SNMP applications.
- Learn how to use a low-level troubleshooting capability in the event that other SNMP applications produce questionable results.
- Poll any OID and verify SNMP agent responses.
- Use stable and reliable CLI commands. UCD-SNMP is unobstructed by GUI functionality.
- Explore and research MIB content.
- Discover what functions are available to manage a Cisco IOS device.
- Create customized scripts and tools.

For this case study, the dial engineers at THENet created a tool called SNMP Commander. The tool aided the MIB research task by enabling dial engineers to build web-based object identification (OIDs) bookmarks, which they could go to without using a keyboard.

By using UCD-SNMP and SNMP Commander, the dial engineers at THENet identified which items the commercial NMS applications would monitor within the network operations center (NOC).

Installing UCD-SNMP and Downloading Cisco MIBs

To install UCD-SNMP and download MIBs from the Cisco FTP site, follow these steps.



Note You can also download individual MIBs from <http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>

-
- Step 1** Go to <http://net-snmp.sourceforge.net/>
- Step 2** Download, compile, and install UCD-SNMP. In this case study, the UCD-SNMP commands are installed in the `/usr/local/bin` directory.
- Step 3** From the Cisco FTP site, download the MIBs into the `/usr/local/share/snmp/mibs` directory on your Solaris workstation. By using the following Unix commands, you can copy the entire bundled v1 MIB tar file from ftp.cisco.com.

```
cd /usr/local/share/snmp/mibs
ftp ftp.cisco.com
cd /pub/mibs/v1
bin
get v1.tar.gz
exit
```

- Step 4** Decompress and untar the files in the `/usr/local/share/snmp/mibs` directory:

```
gzip -d v1.tar.gz
tar -xvf v1.tar
```



Note There are many MIBs in the tar file that you may not use. Regardless, Cisco recommends you keep all the MIBs on file to support your evolving network needs.

Exploring SNMP MIBs for Dial Networks

To explore the MIBs for a Cisco IOS device by using SNMP CLI commands, follow the steps in this section. Poll OID variables by using the commands **snmpget**, **snmpwalk**, and **snmptable**.



Note This section assumes you already have a basic understanding of UCD-SNMP and know how to use its CLI commands.

-
- Step 1** To determine the last restart reason for the router, enter the **snmpget** command and the relevant OID. In the following example, the restart reason is “reload.”

```
onionring:~$ snmpget travis-nas-01.the.net 5urf5h0p .1.3.6.1.4.1.9.2.1.2.0
Counter32 (is a reserved word): At line 6 in /usr/local/share/snmp/mibs/SNMPv2-S
MI-V1SMI.my
Gauge32 (is a reserved word): At line 7 in /usr/local/share/snmp/mibs/SNMPv2-SMI
-V1SMI.my
Integer32 (is a reserved word): At line 8 in /usr/local/share/snmp/mibs/SNMPv2-S
MI-V1SMI.my
Did not find 'mib-2' in module RFC1213-MIB (/usr/local/share/snmp/mibs/IANAifTyp
e-MIB-V1SMI.my)
```

```
enterprises.9.2.1.2.0 = "reload"
```

If SNMP-parsing errors are generated, suppress them by appending **2>/dev/null** to the end of the command. Standard output is tagged as 1. Error output is tagged as 2.

```
onionring:~$ snmpget travis-nas-01.the.net 5urf5h0p .1.3.6.1.4.1.9.2.1.2.0 2> /dev/null
enterprises.9.2.1.2.0 = "reload"
onionring:~$
```



Note If no response is returned by the SNMP agent, allow error messages to print to the screen by removing the **2>/dev/null** argument.

Step 2 Check the system up time by entering the **snmpget** command and sysUpTime OID:

```
onionring:~$ snmpget travis-nas-01.the.net 5urf5h0p .1.3.6.1.2.1.system.sysUpTime.0 2> /dev/null
system.sysUpTime.0 = Timeticks: (45450609) 5 days, 6:15:06.09
onionring:~$
```

Step 3 To gather basic configuration management information about the Cisco IOS device, enter the **snmpwalk** command and the system OID.

```
onionring:~$ snmpwalk travis-nas-01.the.net 5urf5h0p system 2> /dev/null
system.sysDescr.0 = "Cisco Internetwork Operating System Software ..IOS (tm) 5800
Software (C5800-P4-M), Version 12.1(2a)T1, RELEASE SOFTWARE (fc2)..Copyright
(c) 1986-2000 by cisco Systems, Inc...Compiled Mon 12-Jun-00 23:13 by ccai"
system.sysObjectID.0 = OID: enterprises.9.1.188
system.sysUpTime.0 = Timeticks: (45492606) 5 days, 6:22:06.06
system.sysContact.0 = "net-admin@aurora.the.net"
system.sysName.0 = "travis-nas-01.the.net"
system.sysLocation.0 = "Lake Travis (Austin) Dial POP"
system.sysServices.0 = 78
system.8.0 = Timeticks: (0) 0:00:00.00
onionring:~$
```

Step 4 Change the OID environmental prefix by entering the commands **prefix** and **export prefix**. This step reduces the number of key strokes you must enter at the command line.

```
onionring:~$ snmpget travis-nas-01.the.net 5urf5h0p .1.3.6.1.4.1.9.2.1.2.0 2> /dev/null
enterprises.9.2.1.2.0 = "reload"
onionring:~$ PREFIX=.1.3.6.1.4.1.9
onionring:~$ export PREFIX
onionring:~$ snmpget travis-nas-01.the.net 5urf5h0p 2.1.2.0 2> /dev/null
enterprises.9.2.1.2.0 = "reload"
onionring:~$
```

The UCD-SNMP application attaches a prefix to the requested variable unless it is fully qualified (for example, unless the variable starts with a period "."). By default, the prefix points to the MIB-II node .1.3.6.1.2.1 location. The Cisco enterprises prefix points to .1.3.6.1.4.1.9

Step 5 Inspect the IP address entry table by entering the **snmptable** command and ipAddrTable OID:

```
onionring:~$ snmptable travis-nas-01.the.net 5urf5h0p ip.ipAddrTable 2> /dev/null
SNMP table: ip.ipAddrTable.ipAddrEntry
ipAdEntAddr ipAdEntIfIndex ipAdEntNetMask ipAdEntBcastAddr ipAdEntReasmMaxSize
172.21.10.1          351    255.255.255.255          1          18024
172.21.101.20       289    255.255.255.0           1          18024
onionring:~$
```

- Step 6** Poll the interfaces table and redirect the output to a text file by entering the **snmptable** command and ifTable OID:

```
onionring:~$ snmptable travis-nas-01.the.net 5urf5h0p interfaces.ifTable
> /export/home/www/travis-nas-01_ifTable.txt
onionring:~$
```



Note Do not forget the space between > and /export

- Step 7** Inspect the contents of the interfaces table by entering the **cat** command. In the following Cisco AS5800 example, notice the interface descriptions (ifDescr) and types (ifType). There is one PPP and DS0 entry for each serial interface.

```
onionring:~$ cat /export/home/www/travis-nas-01_ifTable.txt
SNMP table: interfaces.ifTable.ifEntry
```

ifIndex	ifDescr	ifType	ifMtu	ifSpeed	
1	"Async1/2/00"	other	1500	9000	
2	"Async1/2/01"	other	1500	9000	
3	"Async1/2/02"	other	1500	9000	
.					
.					
.					
289	"FastEthernet0/0/0"	ethernetCsmacd	1500	100000000	
290	"Null0"	other	1500	4294967295	
291	"T1 1/0/0"	ds1	?	?	
292	"T1 1/0/1"	ds1	?	?	
.					
.					
.					
301	"T1 1/0/10"	ds1	?	?	
302	"T1 1/0/11"	ds1	?	?	
303	"Serial1/0/0:0"	propPointToPointSerial	1500	64000	
304	"Serial1/0/0:1"	propPointToPointSerial	1500	64000	
.					
.					
.					
326	"Serial1/0/0:23"	lapd	1500	64000	
327	"Serial1/0/0:23-Signaling"	isdn	1500	64000	
328	"Serial1/0/0:0-Bearer Channel"	ds0	?	?	
329	"Serial1/0/0:1-Bearer Channel"	ds0	?	?	
.					
.					
.					
350	"Serial1/0/0:22-Bearer Channel"	ds0	?	?	
351	"Loopback0"	softwareLoopback	15144294967295		

About SNMP Commander

The dial engineers at THENet created a tool called SNMP Commander that:

- Provides web-based access to UCD-SNMP CLI commands.
- Builds web-based OID bookmarks, which enable you to go to OIDs without using a keyboard.

- Aids the MIB exploration and NMS design tasks.

By using SNMP Commander and a web browser, you can:

- Create URL links for the network staff and help desk.
- Identify target OIDs you want to graph by using MRTG.
- Inspect thresholds and events to monitor by using other NMS systems.

The following two components work together to create SNMP Commander:

- **snmpcmds.dat**—A comma separated variables file, which includes a list of SNMP CLI commands. This file is read by the snmpcmds.pl script.

The following is the source code for this file:

```
host:/opt/CSC01wt/conf$ ls
devlist.txt                naslist2.txt
lwt.conf                   snmpcmds.dat
mrtg_01.cfg                swatch-nas.0.cfg
mrtg_01.ok                 swatch-nas.cfg
mrtg_cron.sh               swatch-pers.cfg
mrtg_mauui-nas-01_dial.cfg thenet-tools_menu.lbi
mrtg_mauui-nas-01_dial.ok
host:/opt/CSC01wt/conf$ cat snmpcmds.dat
walk system,snmpwalk,.1.3.6.1.2.1.1,RFC1213-MIB,system
walk cisco.lsystem, snmpwalk, .1.3.6.1.4.1.9.2.1,OLD-CISCO-SYS-MIB,lsystem
table entPhysicalTable,snmptable -ib,.1.3.6.1.2.1.47.1.1.1
,ENTITY-MIB,entPhysicalTable
table entLogicalTable,snmptable -ib,.1.3.6.1.2.1.47.1.2.1,ENTITY-MIB,entLogicalTable
walk chassis,snmpwalk,.1.3.6.1.4.1.9.3.6,OLD-CISCO-CHASSIS,chassis
table cardTable,snmptable -ib,.1.3.6.1.4.1.9.3.6.11,OLD-CISCO-CHASSIS,cardTable
table cardIfIndexTable,snmptable
-ib,.1.3.6.1.4.1.9.3.6.13,OLD-CISCO-CHASSIS,cardIfIndexTable
table cpmCPUTotalTable, snmptable -ib, .1.3.6.1.4.1.9.9.109.1.2.1,Cisco-PROCESS-MIB,
cpmCPUTotalTable
table cpmProcessTable, snmptable -ib,
.1.3.6.1.4.1.9.9.109.1.2.1,Cisco-PROCESS-MIB,cpmProcessTable
table ProcessExtTable, snmptable -ib,
.1.3.6.1.4.1.9.9.109.1.2.2,Cisco-PROCESS-MIB,ProcessExtTable
table ciscoMemoryPoolTable, snmptable -ib,
.1.3.6.1.4.1.9.9.48.1.1,CISCO-MEMORY-POOL-MIB, ciscoMemoryPoolTable
table ciscoFlashDeviceTable,snmptable -ib,.1.3.6.1.4.1.9.9.10.1.1.2,CISCO-FLASH-MIB,
ciscoFlashDeviceTable
table ciscoFlashChipTable,snmptable
-ib,.1.3.6.1.4.1.9.9.10.1.1.3.1,CISCO-FLASH-MIB,ciscoFlashChipTable
table ciscoFlashPartitionTable,snmptable
-ib,.1.3.6.1.4.1.9.9.10.1.1.4.1,CISCO-FLASH-MIB,ciscoFlashPartitionTable
table ciscoFlashFileTable,snmptable
-ib,.1.3.6.1.4.1.9.9.10.1.1.4.2.1,CISCO-FLASH-MIB,ciscoFlashFileTable
table ciscoFlashCopyTable,snmptable
-ib,.1.3.6.1.4.1.9.9.10.1.2.1,CISCO-FLASH-MIB,ciscoFlashCopyTable
table ciscoFlashPartitioningTable,snmptable
-ib,.1.3.6.1.4.1.9.9.10.1.2.2,CISCO-FLASH-MIB,ciscoFlashPartitioningTable
table ciscoFlashMiscOpTable,snmptable
-ib,.1.3.6.1.4.1.9.9.10.1.2.3,CISCO-FLASH-MIB,ciscoFlashMiscOpTable
table ciscoImageTable,snmptable
-ib,.1.3.6.1.4.1.9.9.25.1.1,CISCO-IMAGE-MIB,ciscoImageTable
walk lflash,snmpwalk,.1.3.6.1.4.1.9.2.10,OLD-CISCO-FLASH-MIB,lflash
table ciscoEnvMonVoltageStatusTable,snmptable -ib,.1.3.6.1.4.1.9.9.13.1.2
,CISCO-ENVMON-MIB,ciscoEnvMonVoltageStatusTable
table ciscoEnvMonTemperatureStatusTable,snmptable -ib,.1.3.6.1.4.1.9.9.13.1.3
,CISCO-ENVMON-MIB,ciscoEnvMonTemperatureStatusTable
table ciscoEnvMonFanStatusTable,snmptable
-ib,.1.3.6.1.4.1.9.9.13.1.4,CISCO-ENVMON-MIB,ciscoEnvMonFanStatusTable
table ciscoEnvMonSupplyStatusTable,snmptable -ib,.1.3.6.1.4.1.9.9.13.1.5
,CISCO-ENVMON-MIB,ciscoEnvMonSupplyStatusTable
walk
cisco5800HealthMonMIB,snmpwalk,.1.3.6.1.4.1.9.10.28,CISCO-5800-HEALTH-MON-MIB,cisco58
00HealthMonMIB
get ifNumber,snmpget -Os, .1.3.6.1.2.1.2.1.0, IF-MIB, ifNumber
table ifTable, snmptable, .1.3.6.1.2.1.2.2, IF-MIB, ifTable
table lifTable, snmptable -ib,.1.3.6.1.4.1.9.2.2.1,OLD-CISCO-INTERFACES-MIB, lifTable
table dot3StatsTable,snmptable -ib,.1.3.6.1.2.1.10.7.2,RFC1398-MIB,dot3StatsTable
table dsx1ConfigTable,snmptable -ib,.1.3.6.1.2.1.10.18.6,RFC1406-MIB,dsx1ConfigTable
table dsx1CurrentTable,snmptable
-ib,.1.3.6.1.2.1.10.18.7,RFC1406-MIB,dsx1CurrentTable
table dsx1IntervalTable,snmptable
```

```

-ib, .1.3.6.1.2.1.10.18.8, RFC1406-MIB, dsx1IntervalTable
table dsx1TotalTable, snmptable -ib, .1.3.6.1.2.1.10.18.9, RFC1406-MIB, dsx1TotalTable
table dsx1FarEndCurrentTable, snmptable
-ib, .1.3.6.1.2.1.10.18.10, RFC1406-MIB, dsx1FarEndCurrentTable
table dsx1FracTable, snmptable -ib, .1.3.6.1.2.1.10.18.13, RFC1406-MIB, dsx1FracTable
walk ip, snmpwalk, .1.3.6.1.2.1.4, IP-MIB, ip
table ipAddrTable, snmptable -ib, .1.3.6.1.2.1.4.20, IP-MIB, ipAddrTable
table lipAddrTable, snmptable -ib, .1.3.6.1.4.1.9.2.4.1, OLD-CISCO-IP-MIB, lipAddrTable
table ipRouteTable, snmptable -ib, .1.3.6.1.2.1.4.21, IP-MIB, ipRouteTable
table lipRouteTable, snmptable -ib, .1.3.6.1.4.1.9.2.4.2, OLD-CISCO-IP-MIB, lipRouteTable
get ipRoutingDiscards, snmpget -Os, .1.3.6.1.2.1.4.23.0, IP-MIB, ipRoutingDiscards
table ipNetToMediaTable, snmptable -ib, .1.3.6.1.2.1.4.22, IP-MIB, ipNetToMediaTable
walk icmp, snmpwalk, .1.3.6.1.2.1.5, RFC1213-MIB, icmp
walk tcp, snmpwalk, .1.3.6.1.2.1.6, TCP-MIB, tcp
table tcpConnTable, snmptable -ib, .1.3.6.1.2.1.6.13, TCP-MIB, tcpConnTable
walk udp, snmpwalk, .1.3.6.1.2.1.7, UDP-MIB, udp
table udpTable, snmptable -ib, .1.3.6.1.2.1.7.5, UDP-MIB, udpTable
table cQIfTable, snmptable -ib, .1.3.6.1.4.1.9.9.37.1.1, CISCO-QUEUE-MIB, cQIfTable
table cQStatsTable, snmptable -ib, .1.3.6.1.4.1.9.9.37.1.2, CISCO-QUEUE-MIB, cQStatsTable
table rs232PortTable, snmptable -ib, .1.3.6.1.2.1.10.33.2, RS-232-MIB, rs232PortTable
table rs232AsyncPortTable, snmptable -ib, .1.3.6.1.2.1.10.33.3, RS-232-MIB, rs232AsyncPortTable
table rs232SyncPortTable, snmptable -ib, .1.3.6.1.2.1.10.33.4, RS-232-MIB, rs232SyncPortTable
table rs232InSigTable, snmptable -ib, .1.3.6.1.2.1.10.33.5, RS-232-MIB, rs232InSigTable
table rs232OutSigTable, snmptable -ib, .1.3.6.1.2.1.10.33.6, RS-232-MIB, rs232OutSigTable
table ltsLineTable, snmptable -ib, .1.3.6.1.4.1.9.2.9.2, OLD-CISCO-TS-MIB, ltsLineTable
table cdpInterfaceTable, snmptable
-ib, .1.3.6.1.4.1.9.9.23.1.1.1, CISCO-CDP-MIB, cdpInterfaceTable
table cdpCacheTable, snmptable
-ib, .1.3.6.1.4.1.9.9.23.1.2.1, CISCO-CDP-MIB, cdpCacheTable
table isdnBasicRateTable, snmptable -ib, .1.3.6.1.2.1.10.20.1.1.1, ISDN-MIB, isdnBasicRateTable
table isdnBearerTable, snmptable -ib, .1.3.6.1.2.1.10.20.1.2.1, ISDN-MIB, isdnBearerTable
table isdnSignalingTable, snmptable -ib, .1.3.6.1.2.1.10.20.1.3.2, ISDN-MIB, isdnSignalingTable
table isdnSignalingStatsTable, snmptable -ib, .1.3.6.1.2.1.10.20.1.3.3, ISDN-MIB, isdnSignalingStatsTable
table isdnLapdTable, snmptable -ib, .1.3.6.1.2.1.10.20.1.3.4, ISDN-MIB, isdnLapdTable
table isdnEndpointTable, snmptable -ib, .1.3.6.1.2.1.10.20.1.4.2, ISDN-MIB, isdnEndpointTable
table isdnDirectoryTable, snmptable -ib, .1.3.6.1.2.1.10.20.1.5.1, ISDN-MIB, isdnDirectoryTable
table ccasDs1IfCfgTable, snmptable
-ib, .1.3.6.1.4.1.9.9.85.1.1.1, CISCO-CAS-IF-MIB, ccasDs1IfCfgTable
table dialCtlPeer, snmptable -ib, .1.3.6.1.2.1.10.21.1.2.1, DIAL-CONTROL-MIB, dialCtlPeer
table demandNbrTable, snmptable
-ib, .1.3.6.1.4.1.9.9.26.1.1.1, CISCO-ISDN-MIB, demandNbrTable
walk
ciscoCallResourcePoolMIB, snmpwalk, .1.3.6.1.4.1.9.9.124, CISCO-CALL-RESOURCE-POOL-MIB, ciscoCallResourcePoolMIB
table cmmFaxGeneralCfg, snmptable
-ib, .1.3.6.1.4.1.9.9.102.1.4, CISCO-MMAIL-DIAL-CONTROL-MIB, cmmFaxGeneralCfg
table ciscoCallHistoryTable, snmptable
-ib, .1.3.6.1.4.1.9.9.27.1.1.3, CISCO-CALL-HISTORY-MIB, ciscoCallHistoryTable
table cpmDS0UsageTable, snmptable -ib, .1.3.6.1.4.1.9.10.19.1.1.1, CISCO-POP-MGMT-MIB, cpmDS0UsageTable
get ISDNCfgBChanInUseForAnalog, snmpget,
.1.3.6.1.4.1.9.10.19.1.1.2.0, CISCO-POP-MGMT-MIB, cpmISDNCfgBChanInUseForAnalog
get ISDNCfgBChannelsInUse, snmpget, .1.3.6.1.4.1.9.10.19.1.1.3.0, CISCO-POP-MGMT-MIB, cpmISDNCfgBChannelsInUse

```

```

get cpmActiveDSOs, snmpget, .1.3.6.1.4.1.9.10.19.1.1.4.0,CISCO-POP-MGMT-MIB,
cpmActiveDSOs
get cpmPPPCalls, snmpget, .1.3.6.1.4.1.9.10.19.1.1.5.0,CISCO-POP-MGMT-MIB,
cpmPPPCalls
get cpmV120Calls, snmpget, .1.3.6.1.4.1.9.10.19.1.1.6.0,CISCO-POP-MGMT-MIB,
cpmV120Calls
get cpmV110Calls, snmpget, .1.3.6.1.4.1.9.10.19.1.1.7.0,CISCO-POP-MGMT-MIB,
cpmV110Calls
get cpmActiveDSOsHighWaterMark, snmpget,
.1.3.6.1.4.1.9.10.19.1.1.8.0,CISCO-POP-MGMT-MIB, cpmActiveDSOsHighWaterMark
table DS1DS0 Usage Table, snmptable -ib,
.1.3.6.1.4.1.9.10.19.1.1.9,CISCO-POP-MGMT-MIB, cpmDS1DS0UsageTable
get cpmSW56CfgBChannelsInUse, snmpget,
.1.3.6.1.4.1.9.10.19.1.1.10.0,CISCO-POP-MGMT-MIB, cpmSW56CfgBChannelsInUse
walk cpmCallFailure, snmpwalk, .1.3.6.1.4.1.9.10.19.1.2,CISCO-POP-MGMT-MIB,
cpmCallFailure
table cpmActiveCallSummaryTable, snmptable -ib,
.1.3.6.1.4.1.9.10.19.1.3.1,CISCO-POP-MGMT-MIB, cpmActiveCallSummaryTable
get cpmCallHistorySummaryTableMaxLength, snmpget, .1.3.6.1.4.1.9.10.19.1.4.1.0
,CISCO-POP-MGMT-MIB, cpmCallHistorySummaryTableMaxLength
get cpmCallHistorySummaryRetainTimer, snmpget, .1.3.6.1.4.1.9.10.19.1.4.2.0
,CISCO-POP-MGMT-MIB, cpmCallHistorySummaryRetainTimer
table cpmCallHistorySummaryTable, snmptable -ib,
.1.3.6.1.4.1.9.10.19.1.4.3,CISCO-POP-MGMT-MIB, cpmCallHistorySummaryTable
walk cmSystemInfo, snmpwalk, .1.3.6.1.4.1.9.9.47.1.1,CISCO-MODEM-MGMT-MIB,
cmSystemInfo
table cmGroupTable, snmptable -ib, .1.3.6.1.4.1.9.9.47.1.2.1,CISCO-MODEM-MGMT-MIB,
cmGroupTable
table cmGroupMemberTable, snmptable -ib,
.1.3.6.1.4.1.9.9.47.1.2.2,CISCO-MODEM-MGMT-MIB, cmGroupMemberTable
table cmLineStatusTable, snmptable -ib,
.1.3.6.1.4.1.9.9.47.1.3.1,CISCO-MODEM-MGMT-MIB, cmLineStatusTable
table cmLineConfigTable, snmptable -ib,
.1.3.6.1.4.1.9.9.47.1.3.2,CISCO-MODEM-MGMT-MIB, cmLineConfigTable
table cmLineStatisticsTable, snmptable -ib,
.1.3.6.1.4.1.9.9.47.1.3.3,CISCO-MODEM-MGMT-MIB, cmLineStatisticsTable
table cmLineSpeedStatisticsTable, snmptable -ib,
.1.3.6.1.4.1.9.9.47.1.3.4,CISCO-MODEM-MGMT-MIB, cmLineSpeedStatisticsTable
walk CISCO-AAA-SERVER-MIB,snmpwalk,.1.3.6.1.4.1.9.10.56,CISCO-AAA-SERVER-MIB,ls
CISCO-AAA-SERVER-MIB
walk
CISCO-AAA-SESSION-MIB,snmpwalk,.1.3.6.1.4.1.9.9.150,CISCO-AAA-SESSION-MIB,CISCO-AAA-S
ESSION-MIB
walk ciscoAAAServerCapability, snmpwalk,
.1.3.6.1.4.1.9.7.129,CISCO-AAA-SERVER-CAPABILITY, ciscoAAAServerCapability
get cvpdnTunnelTotal,snmpget
-Os,.1.3.6.1.4.1.9.10.24.1.1.1.0,CISCO-VPDN-MGMT-MIB, cvpdnTunnelTotal
get cvpdnSessionTotal, snmpget
-Os,.1.3.6.1.4.1.9.10.24.1.1.2.0,CISCO-VPDN-MGMT-MIB, cvpdnSessionTotal
get cvpdnDeniedUsersTotal,snmpget
-Os,.1.3.6.1.4.1.9.10.24.1.1.3.0,CISCO-VPDN-MGMT-MIB, cvpdnDeniedUsersTotal
table cvpdnSystemTable,snmptable
-ib,.1.3.6.1.4.1.9.10.24.1.1.4,CISCO-VPDN-MGMT-MIB, cvpdnSystemTable
table cvpdnTunnelTable,snmptable
-ib,.1.3.6.1.4.1.9.10.24.1.2.1,CISCO-VPDN-MGMT-MIB, cvpdnTunnelTable
table cvpdnTunnelAttrTable,snmptable
-ib,.1.3.6.1.4.1.9.10.24.1.2.2,CISCO-VPDN-MGMT-MIB, cvpdnTunnelAttrTable
table cvpdnTunnelSessionTable,snmptable
-ib,.1.3.6.1.4.1.9.10.24.1.3.1,CISCO-VPDN-MGMT-MIB, cvpdnTunnelSessionTable
table cvpdncvpdnSessionAttrTable,snmptable
-ib,.1.3.6.1.4.1.9.10.24.1.3.2,CISCO-VPDN-MGMT-MIB, cvpdnSessionAttrTable
table cvpdnUserToFailHistInfoTable, snmptable
-ib,.1.3.6.1.4.1.9.10.24.1.4.1,CISCO-VPDN-MGMT-MIB, cvpdnUserToFailHistInfoTable
table rttMonApplSupportedRttTypesTable, snmptable

```

```

-ib, .1.3.6.1.4.1.9.9.42.1.1.7, CISCO-RTTMON-MIB, rttMonApplSupportedRttTypesTable
table rttMonApplSupportedProtocolsTable, snmptable
-ib, .1.3.6.1.4.1.9.9.42.1.1.8, CISCO-RTTMON-MIB, rttMonApplSupportedProtocolsTable
walk expressionMIB, snmpwalk, .1.3.6.1.4.1.9.10.22, EXPRESSION-MIB, expressionMIB
walk snmpFrameworkMIB, snmpwalk, .1.3.6.1.6.3.10, SNMP-FRAMEWORK-MIB, snmpFrameworkMIB
walk snmp, snmpwalk, .1.3.6.1.2.1.11, SNMPv2-MIB, snmp
walk rmon, snmpwalk, .1.3.6.1.2.1.16, RMON-MIB, rmon
table ccmHistoryEventTable, snmptable
-ib, .1.3.6.1.4.1.9.9.43.1.1.6, CISCO-CONFIG-MAN, ccmHistoryEventTable
walk clogBasic, snmpwalk, .1.3.6.1.4.1.9.9.41.1.1, CISCO-SYSLOG-MIB, clogBasic
table clogHistoryTable, snmptable
-ib, .1.3.6.1.4.1.9.9.41.1.2.3, CISCO-SYSLOG-MIB, clogHistoryTable
walk ciscoFtpClientMIB, snmpwalk, .1.3.6.1.4.1.9.9.80, CISCO-FTP-CLIENT-MIB, ciscoFtpCli

```

- **snmpcmds.pl**—A script that loads and reads a data file. You can use additional data files by creating multiple instances of the original script and altering the data file descriptor.

The following is the source code for this file:

```
host:/opt/CSCOlwt/bin$ ls
device_list.pl  init_lwt.pl      mcr_search.pl    mcr_viewer
dnszone.pl      init_lwt.sh      mcr_sum          snmpcmds.pl
dnszone_dump    loglopper        mcr_view-01     userstat
host:/opt/CSCOlwt/bin$ cat snmpcmds.pl
#!/usr/local/bin/perl -w
# -----
# SNMP Commander.pl v0.3
#
# owner:
# Last Updated:
# -----

# Main program first....
# Using the lightweight CGI Perl Module by ....

use CGI_Lite;

#
# =====PER-SITE VARIABLES=====
#
my $LWTBASE = "/opt/CSCOlwt"; # Base dir for Lightweight Tools.
my $devicelist_fn = "$LWTBASE/conf/devlist.txt"; # Device List File
my $snmpcmds_fn = "$LWTBASE/conf/snmpcmds.dat"; # SNMP Commands File
$stopleft_image = '';
# HTML Component
my $leftmenu_fn = "$LWTBASE/conf/thenet-tools_menu.lbi"; # HTML Component

# add read only Community String

# ===== End of Per-Site Variables =====

$vers = "v0.3";
# Local Variables
my $i = 0; # the universal counter
my $messages = "";

# Create an instance of CGI_Lite to bring in the CGI/Form variables.
#
# Use CGI_lite utilities to create scalars of each CGI AV Pair.
# $cmdreq = "snmp pseudo-code" requested, $agent = selected node.

my $cgi = new CGI_Lite;
my %form = $cgi->parse_form_data;
$cgi->create_variables (\%form);

#Create debug and messaging string
# add to $messages as needed to debug script and functions.

#
# ===== Parse Lightweight Tools files into VARIABLES =====

# Read Files, and set more variables based on them.
# The snmpcmds.dat file is inspected for command definitions. Format by example at this time. No whitespace.
# 1) snmpcmds.dat file is inspected for command definitions.

open (SNMPCMDS, $snmpcmds_fn) || ($messages .= "Can't open SNMP_Commander Data:
file: $snmpcmds_fn<br>\n");
@snmplines = <SNMPCMDS>;
close(SNMPCMDS);
```

```

# prepare SNMP Command Arrays for Commander Interface and Command Execution.
# this routine defines the file input format. 5 fields seperated by "," only.
# i.e. Psuedo-Command,ucd-snmpcmd,numeric_oid,MIB-NAME,symbolic_oid

foreach $snmpline (@snmplines) {
    ($cmd_text[$i], $cmd_verb[$i], $oid[$i], $mib[$i], $oid_symb[$i]) = split
    ("", $snmpline);
    $i++;
}
# remove all leading and trailing whitespace
foreach (@cmd_text, @cmd_verb, @oid, @mib, @oid_symb) {
    s/^\s+//;
    s/\s+$//;
}

# Next, we open and parse the the device_list.txt file.
# The file format is one Device Name per line, no whitespace.

open (DEVLIST, $devicelist_fn) || ($messages .= "Can't open Device List Data: fi
le: $devicelist_fn\n");
@devices = <DEVLIST>;
close(DEVLIST);
chomp @devices;

# ----- Read HTML Template into variables -----

open (LEFTMENU, $leftmenu_fn) || ($messages .= "Can't open Left Menu HTML: file:
$leftmenu_fn\n");
foreach $line (<LEFTMENU>) {$leftmenu .= $line}
close (LEFTMENU);

# ===== Generate Dynamic HTML Components =====

## First prepare parts
# Spin command text into HTML form selection list.

$html_cmdlist = "";
rm <option>s.
$selected = "";
OSTed by last visit
foreach $cmdtext (@cmd_text) {
    if ($cmdtext eq $cmdreq) {$selected = " selected"} else {$selected = ""}
    $html_cmdlist .= "<option$selected>$cmdtext</option>\n";
}

# Spin device names into HTML form selection list.
$html_devlist = "";
rm <option>s.
$selected = "";
OSTed by last visit

foreach $device (@devices) {
    if ($device eq $agent) {$selected = " selected"} else {$selected = ""}
    $html_devlist .= "<option$selected>$device</option>\n";
}

# If SNMP output was requested, go get it from &snmp_commander in HTML format.
if ($cmdreq ne "") {
    for ($i=0; $i < @cmd_text; $i++) {
        if ($cmdreq eq $cmd_text[$i]) {
            if ($agent) {
                $snmp_results = &snmp_commander($agent, $cmd_verb[$i], $oid[$i])
            }
        } else {$messages .= "Command $cmd_text[$i] found, but no agent requ

```

```

ested!\n"}
        $choice = $i;
        $found = "true";
        last;
    }
}
if (!$found) {$messages .= "requested command $cmdreq not found\n"}
} elseif ($agent) {$messages .= "Device $agent requested, but no snmp command sel
ected!\n"}

# ----- Generate Info about last requested Action -----
#

if ($found) {
    $messages .= "Requested Cmd: $cmdreq\n";
    $messages .= "Requested Device: $agent\n";
    $messages .= "MIB: $mib[$choice]\n";
    $messages .= "OID: Numeric = $oid[$choice], Symbolic = $oid_symb[$choice]\n
";
}

# Location to add debug output to $messages
# uncomment or add other lines as needed to check script state
# $messages .= "passed command was !$cmdreq!\n";
# $messages .= "agent passed was !$agent!\n";
# $messages .= "choice was was !$choice!\n";
# $messages .= "mib[choice] = !$mib[$choice]!\n";
# $messages .= "leftmenu is:\n$leftmenu\n";

# =====
# Print HTML Output
# generate device list in HTML

print "content-type: text/html\n\n";
print <<HTML_OUTPUT;
<html>
<head>
<title>SNMP Commander @ The.Net</title>
</head>

<body bgcolor="#FFFFFF" background="/graphics/ss7_dial1.gif">

<table width="650" border="1">
    <tr>
        <td width="90" valign="center" align="center">$topleft_image</td>

        <td><br>
            <font size="-1" face="Verdana, Arial, Helvetica, sans-serif"><i>SNMP Comma
nder Version $vers</i></font>
        </td>
    </tr>
    <tr>
        <td width="90" valign="top">$leftmenu</td>
        <td valign="top">

<h2>SNMP Commander @ The.Net</h2>
<body bgcolor="#FFFFFF">
<form method="GET" enctype="application/x-www-form-urlencoded">
    <p><b> Select SNMP Command * OID: <select name="cmdreq">$html_cmdlist</select>
<br>
        Select SNMP Agent:    <select name="agent">$html_devlist</select>    Go: <i
nput type="submit" value="Submit">
    </b></p>

```

```

</form>
<hr width="650" align="left">
  <table border="1" bgcolor="#FFFF99" width="100%">
    <tr><td width="100%">
      <b>Program messages:</b>
      <pre width="80">$messages</pre>
    </td></tr>
  </table>

  <table border="1" bgcolor="#AAFFFF" width="100%">
    <tr><td width="100%">
      <pre width="80">
        $snmp_results
      </pre>
    </td></tr>
  </table>
  <b>SNMP Commander $vers</b> Contact: <a href=mailto:net-admin\@aurora.the.net>ne
  t-admin\@aurora.the.net</a>
</td>
</tr>
</table>

</body>
</html>

```

HTML_OUTPUT

```

exit 0;
# end of main program
#
# Subroutines
# =====
#
# SNMP Command Execution Subroutine

sub snmp_commander {
my $lagent = $_[0];
my $lverb = $_[1];
my $loid = $_[2];
my $results = "";
my $comm_string = "underdog";

$results .= "<h3>Query was: $lverb -m all $lagent comm-string $loid 2>/dev/null<
/h3>\n";
$results .= `/usr/local/bin/$lverb -m all $lagent $comm_string $loid 2>/dev/null
\n`;

return $results;
}

```

Setting Up SNMP Commander

To set up SNMP Commander, follow these steps:

- Step 1** From CCO, download snmpcmds.dat and snmpcmds.pl
- Step 2** Customize the files for your environment. When you find useful OIDs, enter them in the snmpcmds.dat file and use the web-based form of the script to research the MIBs. The web tool functions like an SNMP OID bookmark.

Step 3 Test SNMP Commander by using a web browser:

- a. Select an SNMP command and OID.
- b. Select an SNMP agent (Cisco IOS device).
- c. Click **Submit**.
- d. Inspect the program and query messages.

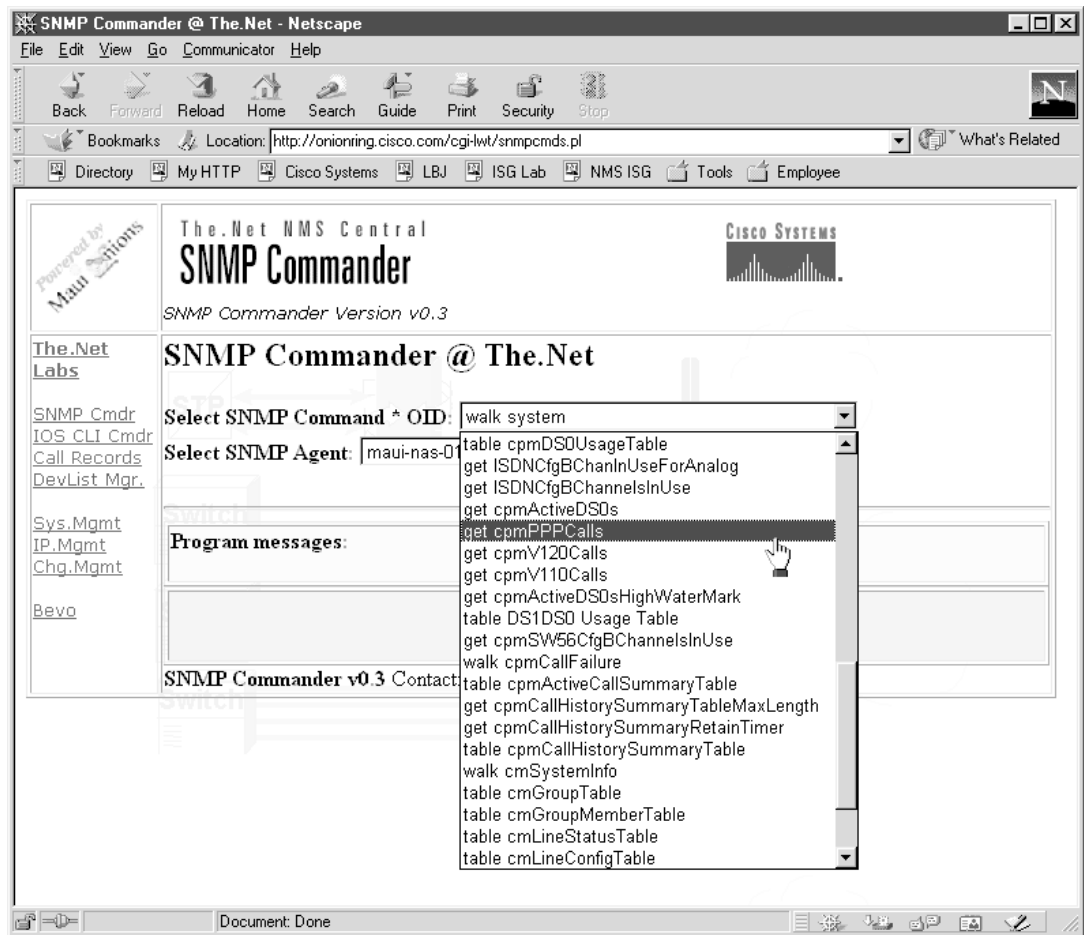
Figure 8 SNMP Commander Tool

Figure 9 Polling Results from the table `cpmActiveCallSummaryTable` Command

The screenshot shows the SNMP Commander web interface in a Netscape browser window. The page title is "The.Net NMS Central SNMP Commander". The interface includes a navigation bar with links like Back, Forward, Reload, Home, Search, Guide, Print, Security, and Stop. The main content area displays the results of a query for the `cpmActiveCallSummaryTable` on the device `maui-nas-01`.

Program messages:

```
Requested Cmd: table cpmActiveCallSummaryTable
Requested Device: maui-nas-01
MIB: CISCO-POP-MGMT-MIB
OID: Numeric = .1.3.6.1.4.1.9.10.19.1.3.1, Symbolic = cpmActiveCallSummaryTable
```

Query was: `snmpget -ib -m all maui-nas-01 comm-string .1.3.6.1.4.1.9.10.19.1.3.1 2>/dev/null`

SNMP table: `enterprises.cisco.ciscoExperiment.ciscoPopMgmtMIB.ciscoPopMgmtMIBObjects.cpmActiveCallSummary.cpmActiveCallSummaryTable`

index	CallStartTimeIndex	CallSummaryIndex	UserID	UserIpAddr	CallType	ModemSlot	ModemPort	CallID
20244644.0	?	?	"rbrown-isdn"	0.0.0.0	digital	-1	-1	0:1:5
20247644.0	?	?	"rbrown-isdn"	0.0.0.0	digital	-1	-1	0:1:5
20786302.0	?	?	"dleyland-isdn"	0.0.0.0	digital	-1	-1	0:0:2

The status bar at the bottom of the browser window shows "Document: Done".

