MG-SOFT Corporation

Network Management Technology Overview

August 2020

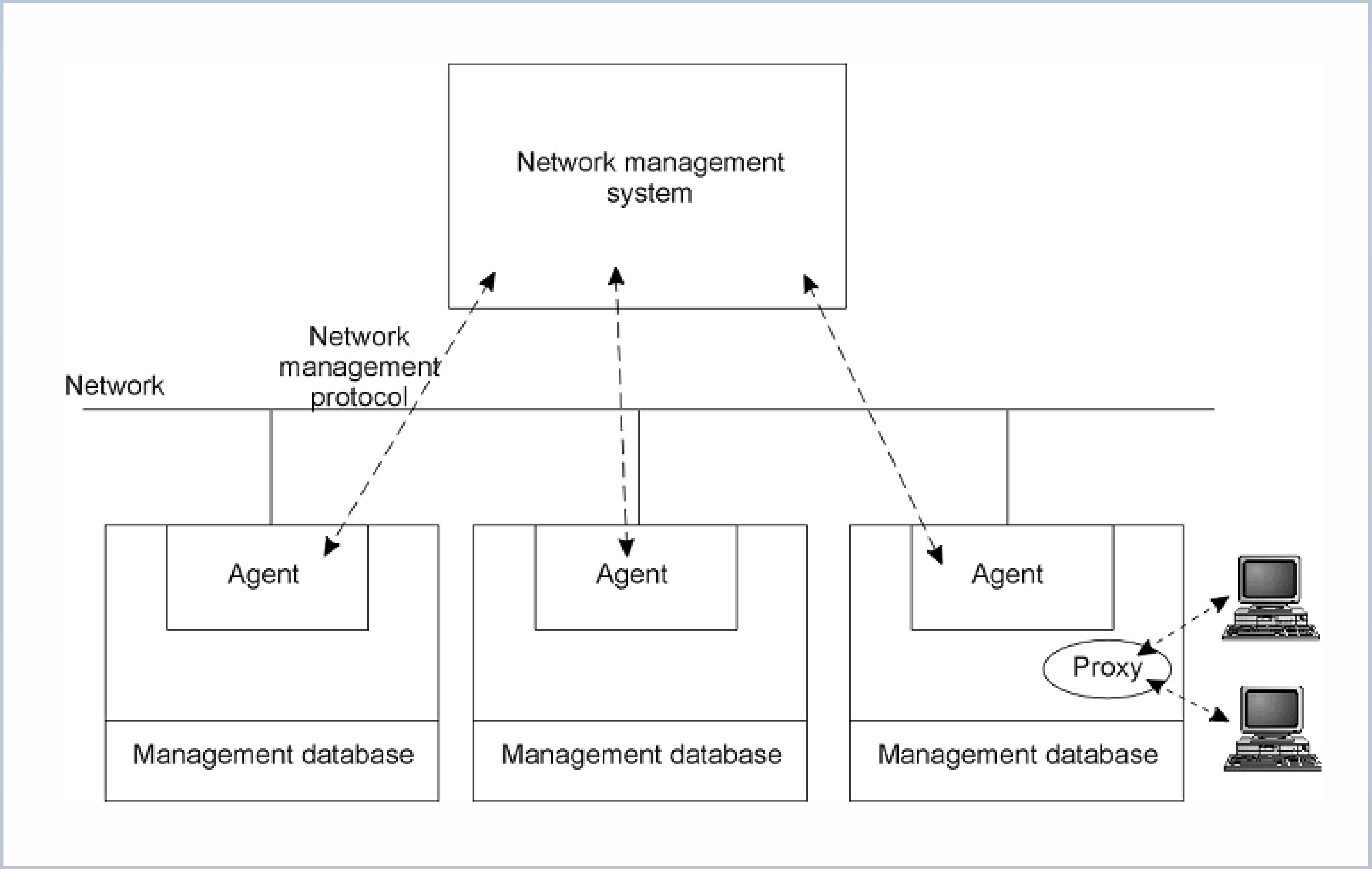
Contents

1. Network Management

What is Network Management?

Network management is the execution of the set of functions required for controlling, planning, allocating, deploying, coordinating, and monitoring the resources of a network.

Network Management (Cont.)



Network Management (Cont.)

OSI Model of Network Management

Management elements (functional areas):

- **F** ault management
- **C** onfiguration management
- **A** ccounting management
- **P** erformance management
- **S** ecurity management

F
C
A
P
S

Network Management
Fault Management

The goal of *fault management* is to detect, log, notify users of, and (to the extent possible) automatically fix network problems to keep the network running effectively.

Fault management is the most widely implemented of the management elements.

Fault Management applications deal with **Events** and **Alarms**.

F
C
A
P
S

Network Management
Fault Management (Cont.)

Fault is a persistent error that requires user attention or action to repair.
Error is a single abnormal occurrence.

Event

An event is an indication of a distinct occurrence that occurred at a specific point in time. Events do not necessarily indicate faults or errors.

Events are triggered by incoming SNMP traps, by status changes detected through polling and by users managing alarms (e.g., acknowledge).

Examples of events include:

- Route change.
- Port status change (e.g. “up” or “down” or “testing”).
- Device becoming reachable by the management station.
- User acknowledgement of an alarm.

Events are written to the database once and **never change**.

Events open, change (e.g., severity, state) and clear alarms.

F
C
A
P
S

Network Management
Fault Management (Cont.)

Alarm

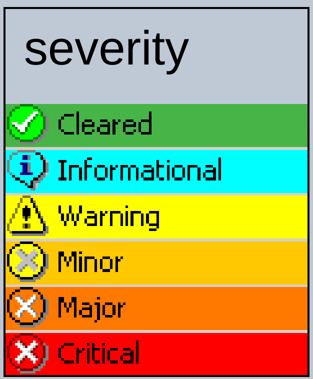
An alarm represents a scenario which involves a fault occurring in the network or management system. Alarms represent the complete fault lifecycle, from the time that the alarm is opened (when the fault is first detected) until it is closed and acknowledged.

Examples of alarms include:

- Link down
- Device unreachable
- HTTP service not responding

An alarm is composed of a sequence of events, each representing a specific point in the alarm lifecycle, e.g:

- Alarm open (e.g., a link-down event), severity: critical
- Alarm clear (e.g., a link-up event), severity: cleared
- Alarm acknowledged by user, state changed to Ack



severity
🟩 Cleared
🟦 Informational
🟨 Warning
🟨 Minor
🟧 Major
🟥 Critical

F
C
A
P
S

Network Management
Fault Management (Cont.)

Elements of Fault Management:

- Monitoring network nodes (polling, notification receiving)
- Generating events/alarms (ITU.T X.733)
- Logging events/alarms
- Autocorrecting when possible
- User notifying, forwarding alarms
- Filtering alarms
- Suppressing and correlating alarms
- Root cause analysis

F
C
A
P
S

Network Management
Configuration Management

The goal of *configuration management* is to monitor network and system configuration information so that the effects on network operation of various versions of hardware and software elements can be tracked and managed.

Different protocols/methods used:

- SNMP
- CLI (vendor-specific)
- TR-069 (DSL domain)
- NETCONF – New!!!

F
C
A
P
S

Network Management Configuration Management (Cont.)

Tasks:

- to gather and store configurations from network devices (this can be done locally or remotely).
- to simplify the configuration of the device
- to track changes that are made to the configuration
- to configure ('provision') circuits or paths through non-switched networks

F
C
A
P
S

Network Management
Accounting Management

The *goal of accounting management* is to gather usage statistics for users of the network services.

Accounting management is concerned with tracking network utilization information, such that individual users, departments, or business units can be appropriately billed or charged for accounting purposes.

RADIUS, TACACS and Diameter are examples of protocols commonly used for accounting.

F
C
A
P
S

Network Management
Accounting Management (Cont.)

For non-billed networks, A in FCAPS is many times interpreted as Administration.

The *goals of administration* are to administer:

- the set of authorized users by establishing users, passwords, and permissions, and
- the operations of the equipment such as by performing software backup and synchronization.

F
C
A
P
S

Network Management
Performance Management

The goal of performance management is to measure and make available various aspects of network performance so that internetwork performance can be maintained at an acceptable level.

The network performance addresses the throughput percentage utilization, error rates, paket loss and response times areas.

Steps:

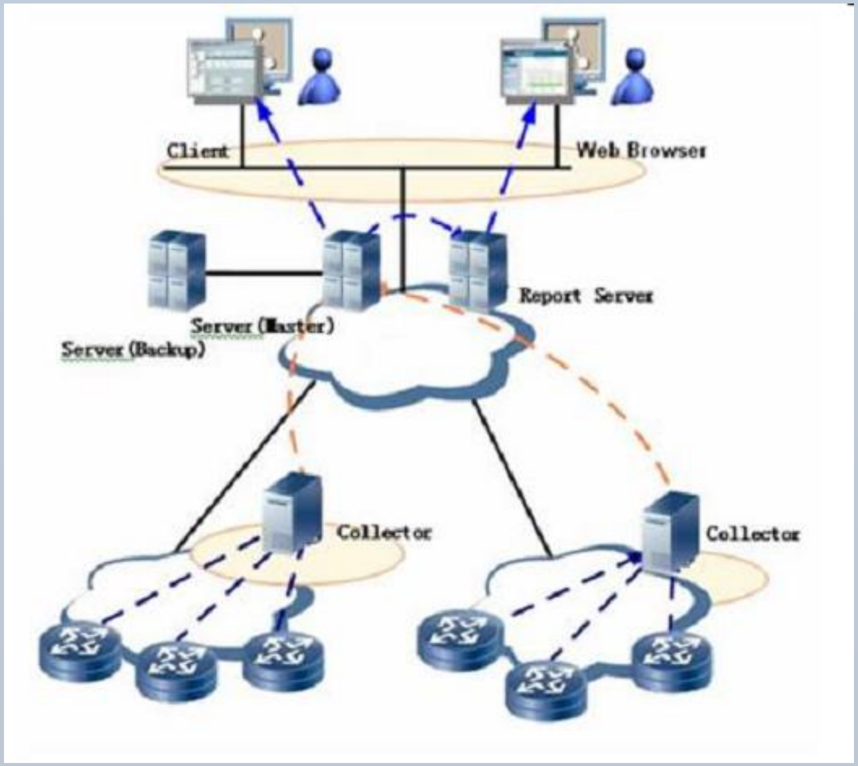
- Data collection (polling, receiving flows)
- Data analyzing (aggregation, averages, trends,...)
- Reporting of alarms if some values cross configurable thresholds.

F
C
A
P
S

Network Management Performance Management (Cont.)

Properties:

- Distributed system
- Storing history data in a database
- Graphical reports
- WEB-based application



F
C
A
P
S

Network Management
Security Management

The goal of security management is to control access to network resources according to local guidelines so that the network cannot be sabotaged (intentionally or unintentionally) and sensitive information cannot be accessed by those without appropriate authorization.

Elements:

- Management of security services
- Logging of security related events
- Management of cryptographic keys
- Access control
- Intrusion detection

Network Management (Cont.)

Network management protocols:

- SNMP (Simple Network Management Protocol)
- CMIP (Common Management Information Protocol)
- WBEM (Web Based Enterprise Management)
- NETCONF (Network Configuration Management)
- Custom protocols

2. SNMP Framework

SNMP framework for managing TCP/IP- based networks consists of:

-MIB (Management Information Base)

MIB modules define the management information (objects) that can be managed via the SNMP. MIB modules are written in language defined in SMI.

-SMI (Structure of Management Information)

Specification of the formal language (adapted subset of ASN.1) for defining management information and the definition of the top-level structure of the OID tree.

-SNMP (Simple Network Management Protocol)

Protocol used to transfer management information between the SNMP entities (agents and managers).

SNMP Framework (Cont.)

3 versions of SNMP management framework:

SNMPv1 Framework:

- **SMIv1** [RFC 1155],
- **MIB-1** [RFC 1156], **MIB-2** [RFC 1213], **Traps** [RFC 1215],
- **SNMPv1** [RFC 1157].

SNMPv2 Framework:

- **SMIv2** [RFC 2578-2580],
- **SNMPv2-MIB** [RFC 1907],
- **SNMPv2c** [RFC 1901,1905-1906].

SNMPv3 Framework:

- **SNMPv3** [RFC 3410-3417], uses SMIv2.

Co-existence between different versions [RFC 3584]

SNMP - Simple Network Management Protocol

Short History

- 1988: Start of SNMPv1
- 1990: SNMPv1 Full Standard
- SNMPv2 Unsuccessful (Resulted in SNMPv2c in 1996)
- 1998: Start of SNMPv3
- 2002: SNMPv3 Full Standard
- 2009: First RFCs for ISMS published
- 2010: (D)TLS for SNMP (RFC 5953)

MIB - Management Information Base

What is a MIB?

A MIB (Management Information Base) contains a definition of management information (MIB objects) that can be collected and controlled by the management application using the SNMP.

MIB objects' properties:

• Name	(human-readable name, e.g., sysUpTime),
• OID	(numerical name - object identifier, e.g.,1.3.6.1.2.1.1.3.0),
• Syntax	(data type, e.g., Octets, Integer, Counter32,...),
• Limits	(valid values, ranges, size (string),....)
• Access	(e.g., read-only, read-write, read-create,...),
• Status	(validity of the object, e.g., current, obsolete,...)
• Value interpretation	(how to format data, display-hint)
• Description	(textual description of the object's meaning)

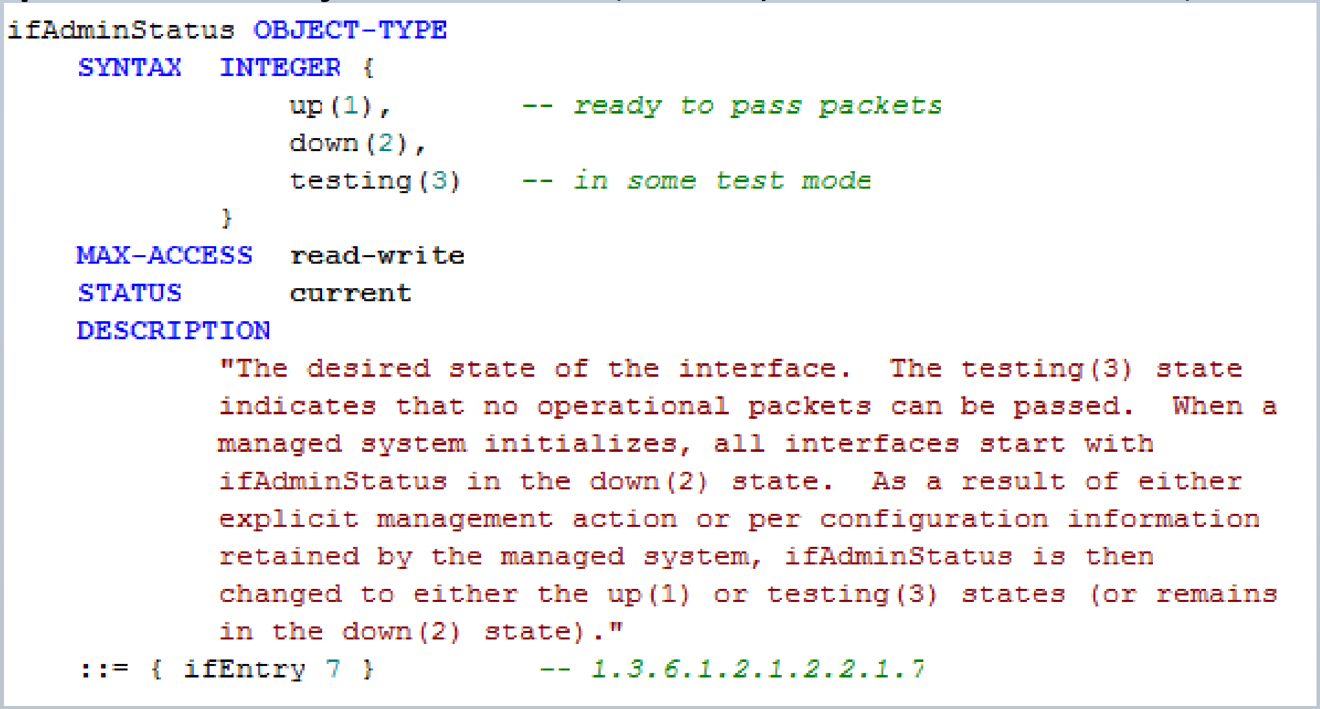
MIB - Management Information Base (Cont.)

MIB module basic structure:

```
<Mib-module-name>  DEFINITIONS ::= BEGIN

        IMPORTS statements
        <MIB object definitions>
END
```

Example of a MIB object definition (SMIv2 syntax, taken from IF-MIB):



```
ifAdminStatus OBJECT-TYPE
    SYNTAX   INTEGER {
                 up(1),        -- ready to pass packets
                 down(2),
                 testing(3)    -- in some test mode
             }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
            "The desired state of the interface.  The testing(3) state
            indicates that no operational packets can be passed.  When a
            managed system initializes, all interfaces start with
            ifAdminStatus in the down(2) state.  As a result of either
            explicit management action or per configuration information
            retained by the managed system, ifAdminStatus is then
            changed to either the up(1) or testing(3) states (or remains
            in the down(2) state)."
    ::= { ifEntry 7 }          -- 1.3.6.1.2.1.2.2.1.7
```

MIB / SMI
Object Identifier (OID)

What is OID?

Object identifier is an ordered sequence of non-negative integers written from left to right, usually separated with dots (.), e.g.:

1.3.6.1.2.1.1.3

In SNMP, OID is an identifier used to uniquely name a MIB object. All MIB objects have administratively assigned OIDs (e.g. OID value assigned to the *sysUpTime* object is *1.3.6.1.2.1.1.3*).

OIDs are hierarchical in nature. They can be presented as nodes in a hierarchically arranged tree structure, called
MIB tree.

MIB / SMI (Cont.)

The top-level structure of the global MIB tree is defined by SMI:

The **root has no label**, and has 3 children:

ccitt(0):	For ITU (formerly the CCITT) standards. (Also seen as *itu(0)*).
iso(1):	For ISO standards.
joint-iso-ccitt(2):	For joint standards. (Also seen as *joint-iso-itu(2)*).

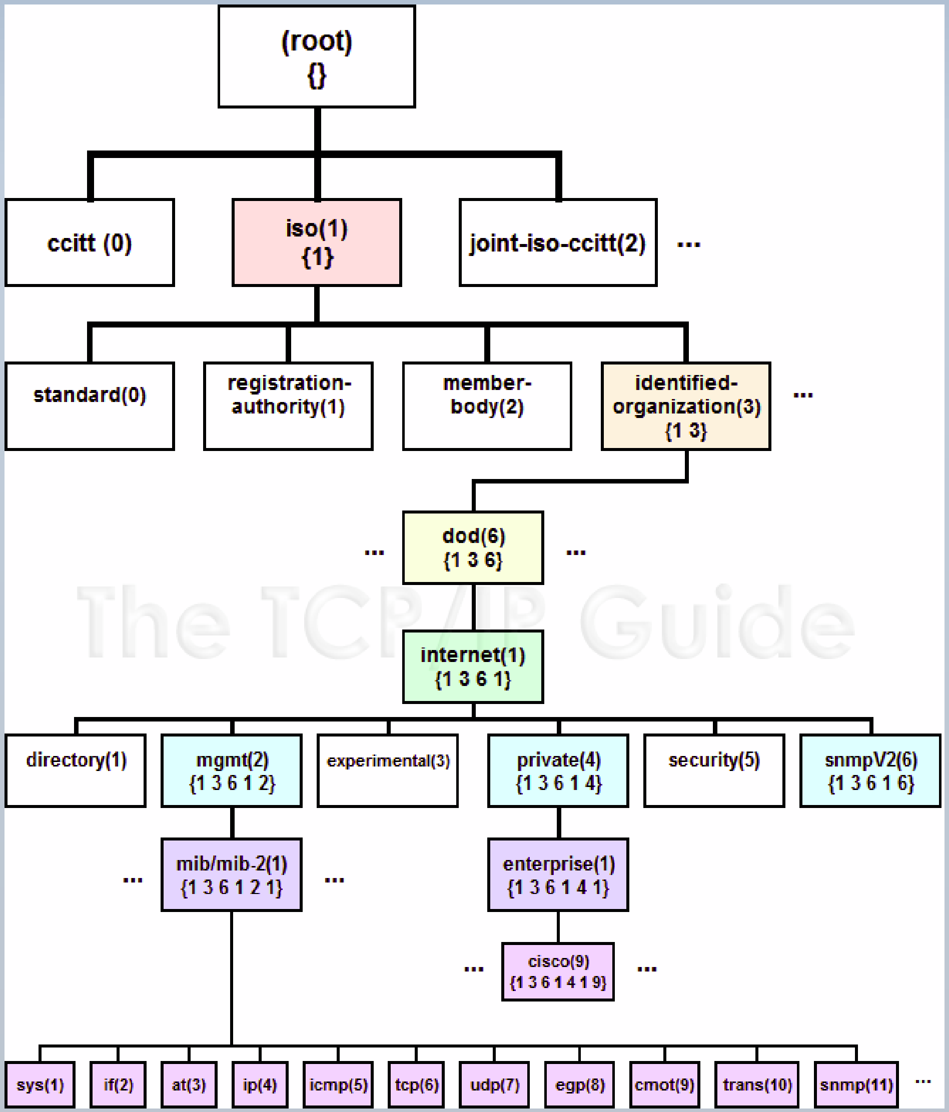
- Within *iso(1)*, the ISO has created a subtree for use by other organizations, called ***org(3)***.

- Within *org(3)*, there is a subtree for the US Department of Defense: ***dod(6)***.

- Within *dod(6)*, there is a subtree called ***internet(1).***

Majority of MIB objects exist under the ***internet*** subtree: ***1.3.6.1*** *or* ***iso.org.dod.internet*** Within this part of the name space, there are six subtrees below:

MIB / SMI (Cont.)



Majority of MIB objects exist under the ***internet*** subtree: ***1.3.6.1*** *or* ***iso.org.dod.internet*** Within this part of the name space, there are six subtrees below:

directory(1): Reserved for future use by ISO.

mgmt(2): The primary subtree where MIB objects are located. This is "1.3.6.1.2". It contains a subtree called *mib-2(1),* which is 1.3.6.1.2.1.

experimental(3): Contains objects used for standards under development. This is "1.3.6.1.3".

private(4): Used for objects defined by **private organizations**. This node, 1.3.6.1.4, has a subtree called *enterprise(1)*, which is **1.3.6.1.4.1**.

security(5): Reserved for security use.

snmpV2(6): Defines objects used specifically for SNMP version 2.

MIB / SMI (Cont.)

Only ***leaf*** objects (OBJECT-TYPE that have no subordinate objects) are accessible via SNMP.

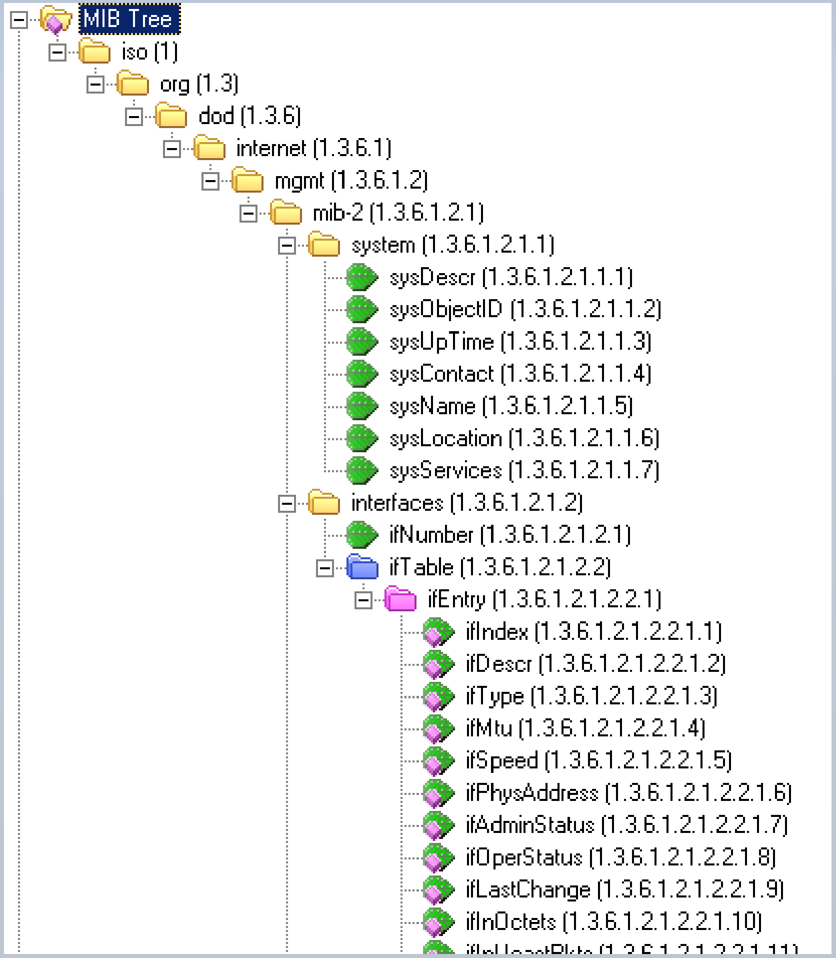
A leaf object can be:

- Scalar object (only one instance)

  e.g.: sysUpTime.0 = 1.3.6.1.2.1.1.3.0

- Columnar object (multiple instances)

  e.g.: ifType.1 = 1.3.6.1.2.1.2.2.1.3.1
  ifType.2 = 1.3.6.1.2.1.2.2.1.3.2
  ifType.3 = 1.3.6.1.2.1.2.2.1.3.3
  etc.

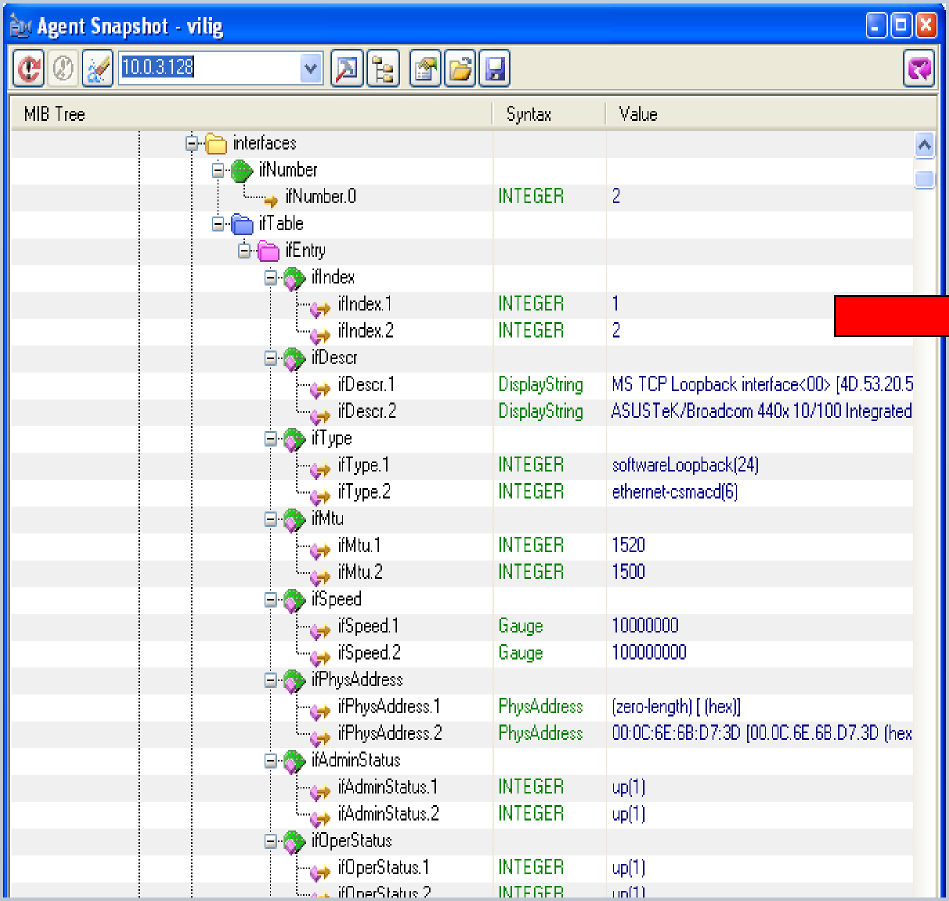


```
MIB Tree
  iso (1)
    org (1.3)
      dod (1.3.6)
        internet (1.3.6.1)
          mgmt (1.3.6.1.2)
            mib-2 (1.3.6.1.2.1)
              system (1.3.6.1.2.1.1)
                sysDescr (1.3.6.1.2.1.1.1)
                sysObjectID (1.3.6.1.2.1.1.2)
                sysUpTime (1.3.6.1.2.1.1.3)
                sysContact (1.3.6.1.2.1.1.4)
                sysName (1.3.6.1.2.1.1.5)
                sysLocation (1.3.6.1.2.1.1.6)
                sysServices (1.3.6.1.2.1.1.7)
              interfaces (1.3.6.1.2.1.2)
                ifNumber (1.3.6.1.2.1.2.1)
                ifTable (1.3.6.1.2.1.2.2)
                  ifEntry (1.3.6.1.2.1.2.2.1)
                    ifIndex (1.3.6.1.2.1.2.2.1.1)
                    ifDescr (1.3.6.1.2.1.2.2.1.2)
                    ifType (1.3.6.1.2.1.2.2.1.3)
                    ifMtu (1.3.6.1.2.1.2.2.1.4)
                    ifSpeed (1.3.6.1.2.1.2.2.1.5)
                    ifPhysAddress (1.3.6.1.2.1.2.2.1.6)
                    ifAdminStatus (1.3.6.1.2.1.2.2.1.7)
                    ifOperStatus (1.3.6.1.2.1.2.2.1.8)
                    ifLastChange (1.3.6.1.2.1.2.2.1.9)
                    ifInOctets (1.3.6.1.2.1.2.2.1.10)
                    ifInUcastPkts (1.3.6.1.2.1.2.2.1.11)
```

MIB (Cont.)

(Conceptual) SNMP Tables

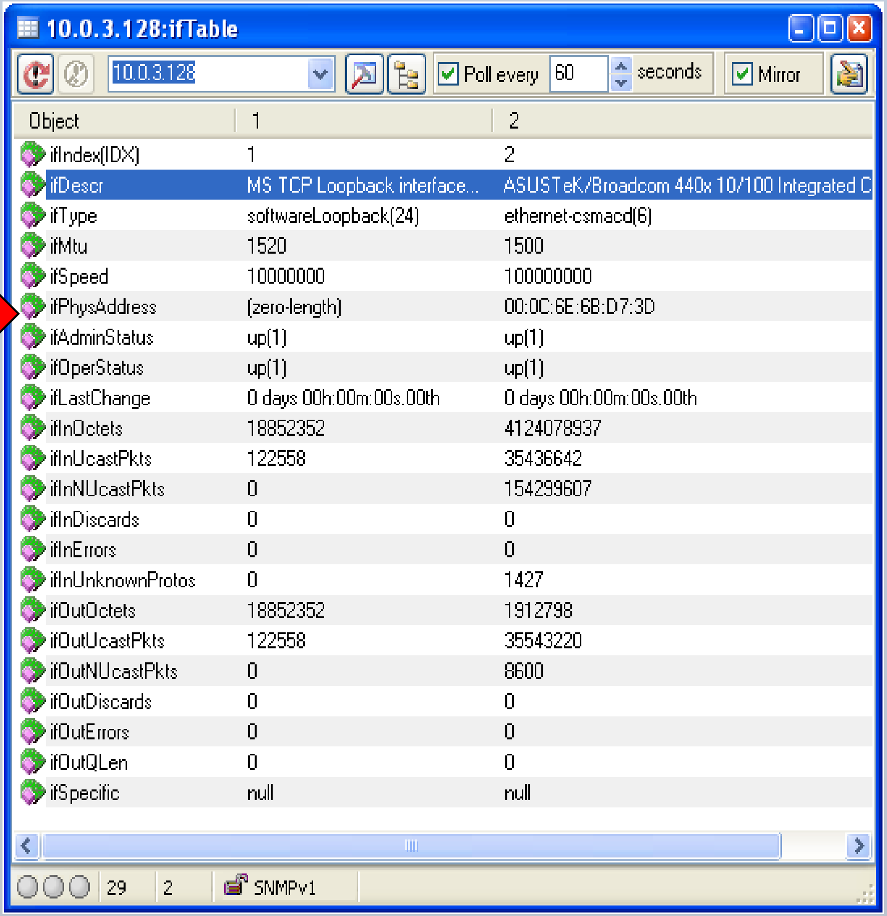
Columnar object instance values can be displayed in form of a table.



Agent Snapshot - vilig

10.0.3.128

MIB Tree	Syntax	Value
interfaces		
ifNumber		
ifNumber.0	INTEGER	2
ifTable		
ifEntry		
ifIndex		
ifIndex.1	INTEGER	1
ifIndex.2	INTEGER	2
ifDescr		
ifDescr.1	DisplayString	MS TCP Loopback interface<00> [4D.53.20.5
ifDescr.2	DisplayString	ASUSTeK/Broadcom 440x 10/100 Integrated
ifType		
ifType.1	INTEGER	softwareLoopback(24)
ifType.2	INTEGER	ethernet-csmacd(6)
ifMtu		
ifMtu.1	INTEGER	1520
ifMtu.2	INTEGER	1500
ifSpeed		
ifSpeed.1	Gauge	10000000
ifSpeed.2	Gauge	100000000
ifPhysAddress		
ifPhysAddress.1	PhysAddress	(zero-length) [(hex)]
ifPhysAddress.2	PhysAddress	00:0C:6E:6B:D7:3D [00.0C.6E.6B.D7.3D (hex
ifAdminStatus		
ifAdminStatus.1	INTEGER	up(1)
ifAdminStatus.2	INTEGER	up(1)
ifOperStatus		
ifOperStatus.1	INTEGER	up(1)
ifOperStatus.2	INTEGER	up(1)



10.0.3.128:ifTable

10.0.3.128 ✓ Poll every 60 seconds ✓ Mirror

Object	1	2
ifIndex(IDX)	1	2
ifDescr	MS TCP Loopback interface...	ASUSTeK/Broadcom 440x 10/100 Integrated C
ifType	softwareLoopback(24)	ethernet-csmacd(6)
ifMtu	1520	1500
ifSpeed	10000000	100000000
ifPhysAddress	(zero-length)	00:0C:6E:6B:D7:3D
ifAdminStatus	up(1)	up(1)
ifOperStatus	up(1)	up(1)
ifLastChange	0 days 00h:00m:00s.00th	0 days 00h:00m:00s.00th
ifInOctets	18852352	4124078937
ifInUcastPkts	122558	35436642
ifInNUcastPkts	0	154299607
ifInDiscards	0	0
ifInErrors	0	0
ifInUnknownProtos	0	1427
ifOutOctets	18852352	1912798
ifOutUcastPkts	122558	35543220
ifOutNUcastPkts	0	8600
ifOutDiscards	0	0
ifOutErrors	0	0
ifOutQLen	0	0
ifSpecific	null	null

29 2 SNMPv1

MIB (Cont.)

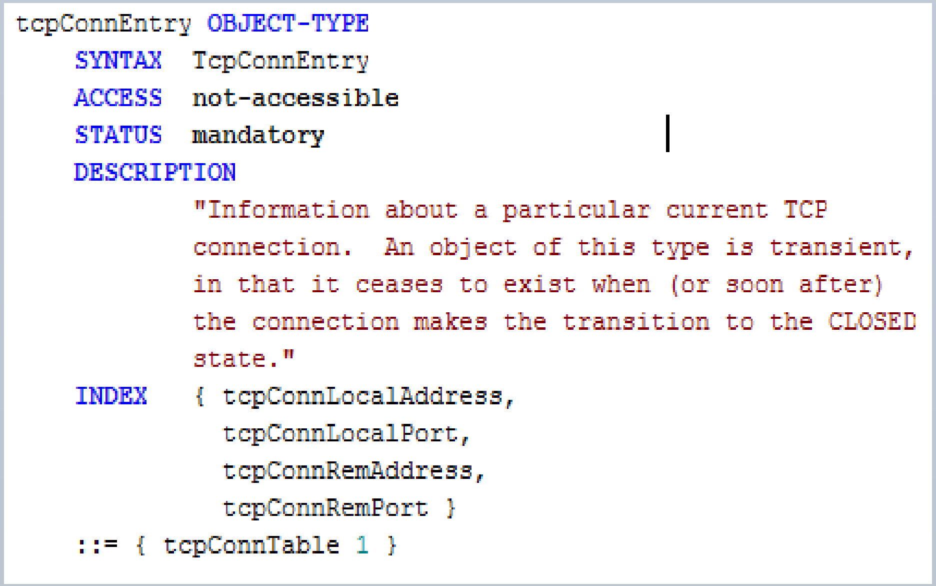
(Conceptual) SNMP Tables – Indexing mechanism

A *row object* defines the *indexing scheme* for the table either by:

- using one or more columns from the same table or
- by using one or more columns from another table.

The first approach involves use of the INDEX clause, while the second approach employs either the INDEX or the AUGMENTS clause (SMIv2).
The **INDEX** or **AUGMENTS** clause determines how rows (or more precisely, instances of columnar objects) in the table are identified.

Example of INDEX definition (from tcpConnTable - 4 INDEX items):

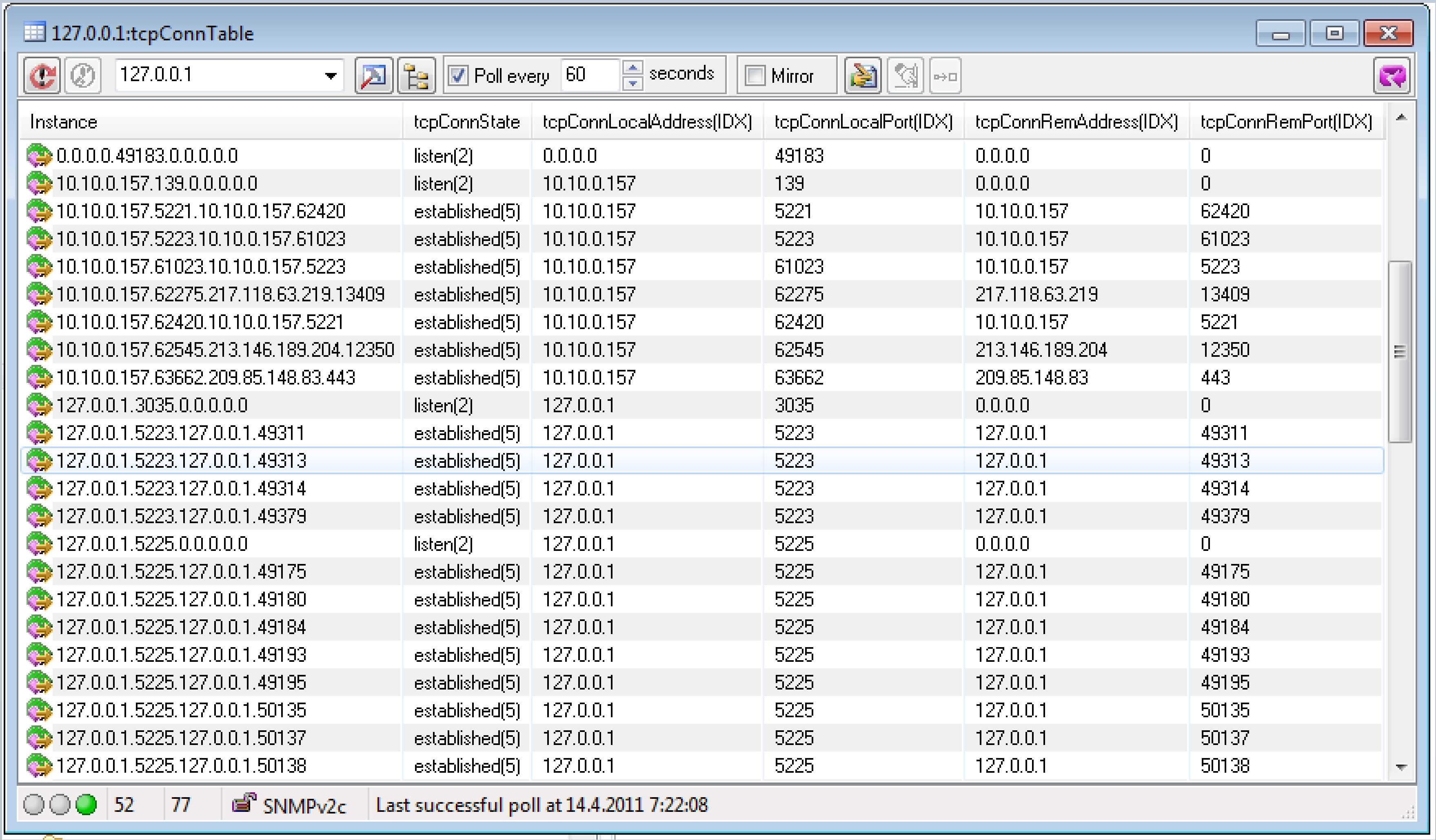


```
tcpConnEntry OBJECT-TYPE
    SYNTAX   TcpConnEntry
    ACCESS   not-accessible
    STATUS   mandatory
    DESCRIPTION
            "Information about a particular current TCP
            connection.  An object of this type is transient,
            in that it ceases to exist when (or soon after)
            the connection makes the transition to the CLOSED
            state."
    INDEX    { tcpConnLocalAddress,
              tcpConnLocalPort,
              tcpConnRemAddress,
              tcpConnRemPort }
    ::= { tcpConnTable 1 }
```

MIB (Cont.)

(Conceptual) SNMP Tables – Indexing mechanism (Cont.)

Example: Viewing the tcpConnTable in MG-SOFT MIB Browser - 4 INDEX (IDX) items:



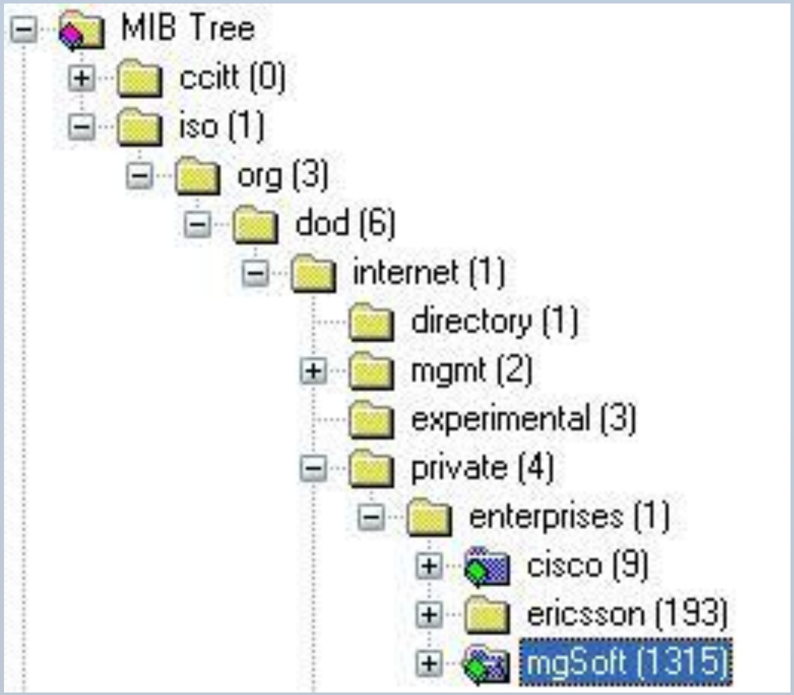
Instance	tcpConnState	tcpConnLocalAddress(IDX)	tcpConnLocalPort(IDX)	tcpConnRemAddress(IDX)	tcpConnRemPort(IDX)
0.0.0.0.49183.0.0.0.0.0	listen(2)	0.0.0.0	49183	0.0.0.0	0
10.10.0.157.139.0.0.0.0.0	listen(2)	10.10.0.157	139	0.0.0.0	0
10.10.0.157.5221.10.10.0.157.62420	established(5)	10.10.0.157	5221	10.10.0.157	62420
10.10.0.157.5223.10.10.0.157.61023	established(5)	10.10.0.157	5223	10.10.0.157	61023
10.10.0.157.61023.10.10.0.157.5223	established(5)	10.10.0.157	61023	10.10.0.157	5223
10.10.0.157.62275.217.118.63.219.13409	established(5)	10.10.0.157	62275	217.118.63.219	13409
10.10.0.157.62420.10.10.0.157.5221	established(5)	10.10.0.157	62420	10.10.0.157	5221
10.10.0.157.62545.213.146.189.204.12350	established(5)	10.10.0.157	62545	213.146.189.204	12350
10.10.0.157.63662.209.85.148.83.443	established(5)	10.10.0.157	63662	209.85.148.83	443
127.0.0.1.3035.0.0.0.0.0	listen(2)	127.0.0.1	3035	0.0.0.0	0
127.0.0.1.5223.127.0.0.1.49311	established(5)	127.0.0.1	5223	127.0.0.1	49311
127.0.0.1.5223.127.0.0.1.49313	established(5)	127.0.0.1	5223	127.0.0.1	49313
127.0.0.1.5223.127.0.0.1.49314	established(5)	127.0.0.1	5223	127.0.0.1	49314
127.0.0.1.5223.127.0.0.1.49379	established(5)	127.0.0.1	5223	127.0.0.1	49379
127.0.0.1.5225.0.0.0.0.0	listen(2)	127.0.0.1	5225	0.0.0.0	0
127.0.0.1.5225.127.0.0.1.49175	established(5)	127.0.0.1	5225	127.0.0.1	49175
127.0.0.1.5225.127.0.0.1.49180	established(5)	127.0.0.1	5225	127.0.0.1	49180
127.0.0.1.5225.127.0.0.1.49184	established(5)	127.0.0.1	5225	127.0.0.1	49184
127.0.0.1.5225.127.0.0.1.49193	established(5)	127.0.0.1	5225	127.0.0.1	49193
127.0.0.1.5225.127.0.0.1.49195	established(5)	127.0.0.1	5225	127.0.0.1	49195
127.0.0.1.5225.127.0.0.1.50135	established(5)	127.0.0.1	5225	127.0.0.1	50135
127.0.0.1.5225.127.0.0.1.50137	established(5)	127.0.0.1	5225	127.0.0.1	50137
127.0.0.1.5225.127.0.0.1.50138	established(5)	127.0.0.1	5225	127.0.0.1	50138

52 77 SNMPv2c Last successful poll at 14.4.2011 7:22:08

MIB (Cont.)

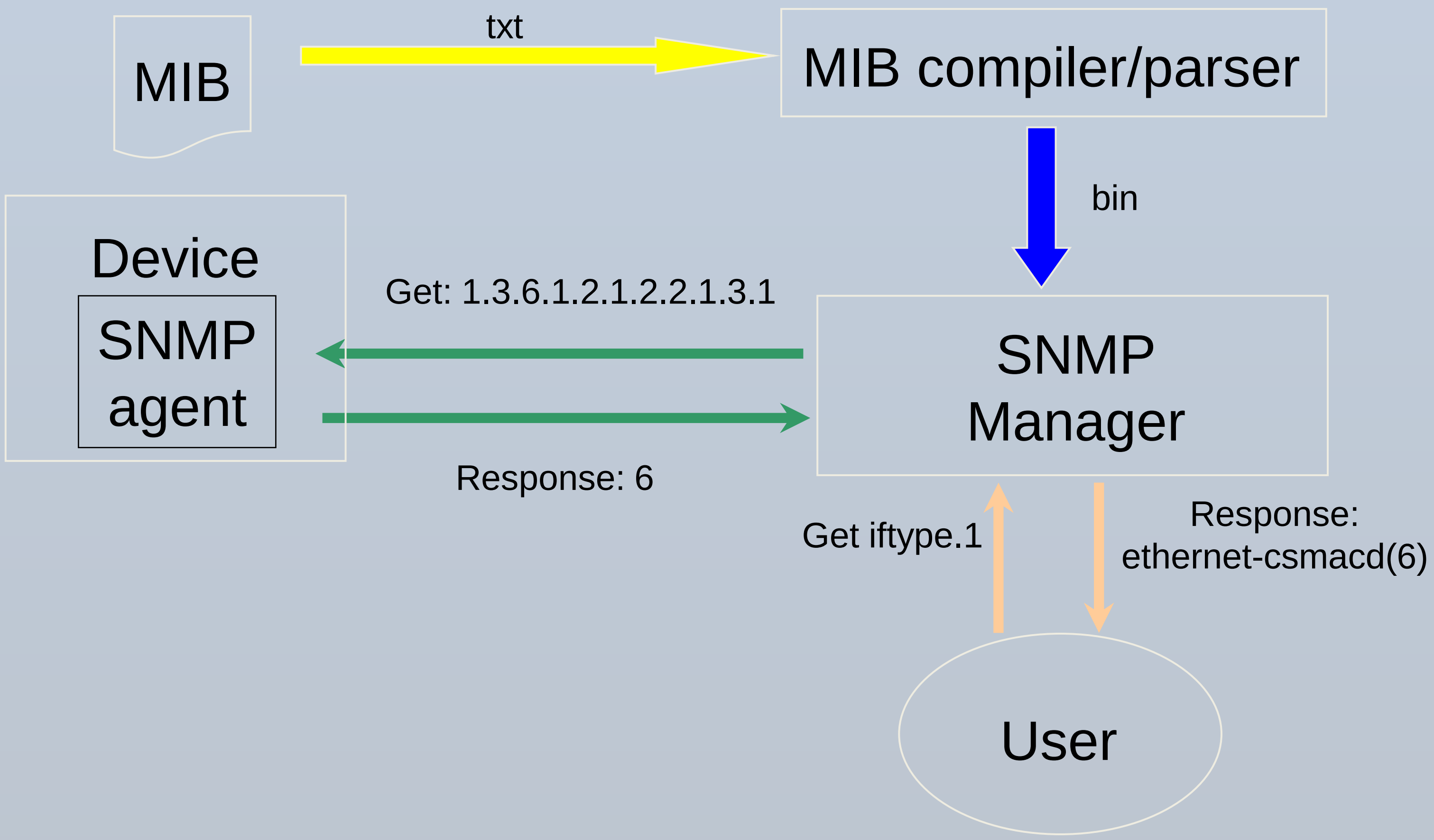
Enterprise MIBs:

- Under private enterprises OID (1.3.6.1.4.1.X)
- Apply for private enterprise number at IANA (Internet Assigned Numbers Authority: http://www.iana.org/assignments/enterprise-numbers)



MIB (Cont.)

Usage of MIBs



MIB (Cont.)

Examples of standard MIBs:

MIB	Description
RFC1213-MIB (also called MIB-2)	Information about system, interfaces, traffic per different protocols (IP, ICMP, UDP, ..)
HOST-RESOURCES-MIB	Information about system resources: processors, memory, disks, processes, installed software ...
BRIDGE-MIB	Bridging information, network topology ...
PRINTER-MIB	Printer information: media (size, capacity), marker supplies, errors, ...
SNMP-USER-BASED-SM-MIB	SNMPv3 User-Based Security model MIB (allows remote management of SNMPv3 users via SNMP)

Simple Network Management Protocol (SNMP)

Basic Facts

What is SNMP?

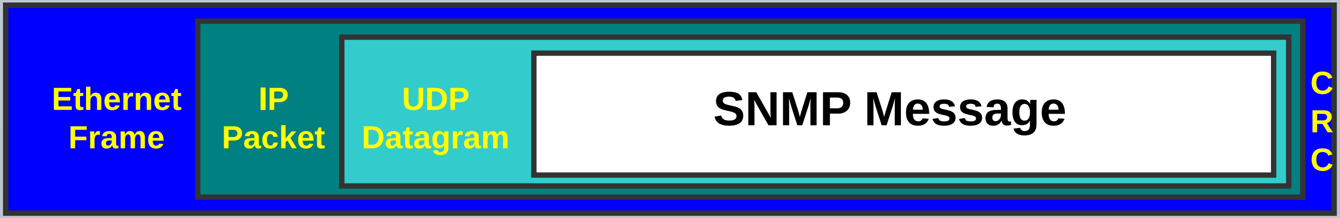
- SNMP is a protocol that allows for remote management of devices (nodes) on the network including servers, workstations, routers, switches, printers, and other managed devices.
- It defines the format and the methods of management communications – exchange of SNMP messages between SNMP entities.

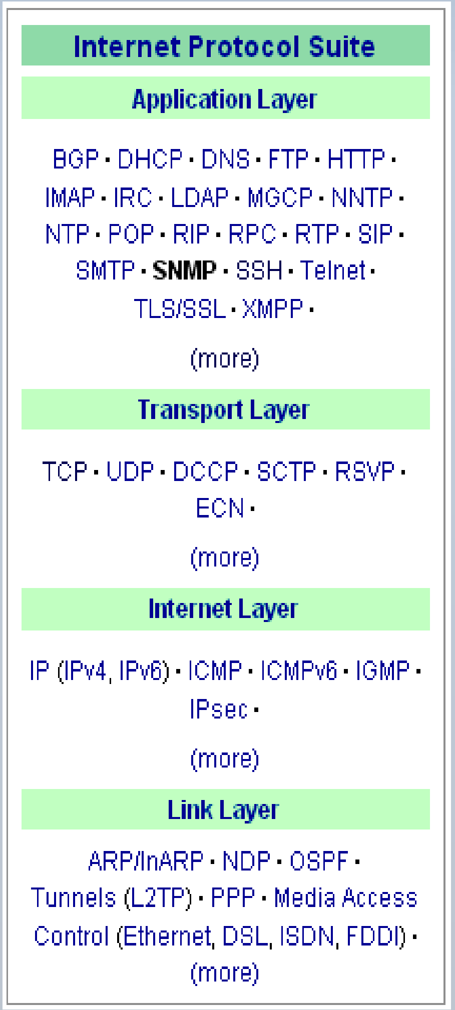
SNMP entities:

- **SNMP Agent** - process running on each managed node collecting information about the device it is running on and exposing it to managers via SNMP.
- **SNMP Manager** - process running on a management computer that requests or alters information on network nodes and receives event notifications from them.

SNMP - Basic Facts (Cont.)

- SNMP is a part of TCP/IP suite of protocols (application layer)
- Transport protocol is UDP over IPv4/IPv6 (connectionless).
- Well-known ports:
  - UDP 161: agents listen for SNMP requests
  - UDP 162: managers listen for SNMP notifications
- Each SNMP message is serialized using ASN.1 BER encoding onto one UDP datagram.





Internet Protocol Suite

Application Layer

BGP · DHCP · DNS · FTP · HTTP · IMAP · IRC · LDAP · MGCP · NNTP · NTP · POP · RIP · RPC · RTP · SIP · SMTP · **SNMP** · SSH · Telnet · TLS/SSL · XMPP ·

(more)

Transport Layer

TCP · UDP · DCCP · SCTP · RSVP · ECN ·

(more)

Internet Layer

IP (IPv4, IPv6) · ICMP · ICMPv6 · IGMP · IPsec ·

(more)

Link Layer

ARP/InARP · NDP · OSPF · Tunnels (L2TP) · PPP · Media Access Control (Ethernet, DSL, ISDN, FDDI) ·

(more)

SNMP - Other Transport Protocols

Several other transport protocol mappings (all use BER encoding):

– SNMP over TCP (experimental)

– SNMP over Novell IPX

– SNMP over AppleTalk (DDP)

– SNMP over OSI (CLTS)

SNMP over TCP (non-standard)

(example of a single SNMP Get operation)

```
Packet  Management                                 Managed
Number  Station                                    Object
             Connection Open...
  1          >--<CTL=SYN>---------------------->
  2          <--<CTL=SYN,ACK>------------------<
  3          >--<CTL=ACK>---------------------->
               Connection now open,
               SNMP Request is sent.
  4          >--<DATA=SNMP Request>------------>
               Response comes back
  5          <--<DATA=SNMP Response, CTL=ACK>---<
  6          >--<CTL=ACK>---------------------->
               Operation is complete,
               Management station initiates the
               close.
  7          >--<CTL=FIN,ACK>------------------>
  8          <--<CTL=ACK>----------------------<
  9          <--<CTL=FIN,ACK>------------------<
 10          >--<CTL=ACK>---------------------->
              Wait 2 MSL
              Connection now closed.
```

SNMP - Operations and Messages

SNMP operations occur through exchange of SNMP messages.

One operation requires the exchange of two messages (e.g., Get-Response), except for “unconfirmed” event report operation (Trap), which requires one message.

SNMP Operations:

1) Retrieval (read) operations:	**Get, GetNext, GetBulk**
2) Modification (write) operation:	**Set**
3) Event reporting operations:	**Trap and Inform**

SNMP Message: A basic unit of SNMP protocol communication in strictly specified format. It consists of the message header and PDU (protocol data unit). Serialized (binary) SNMP messages are sent on the network.

SNMP - Operations by Protocol Version

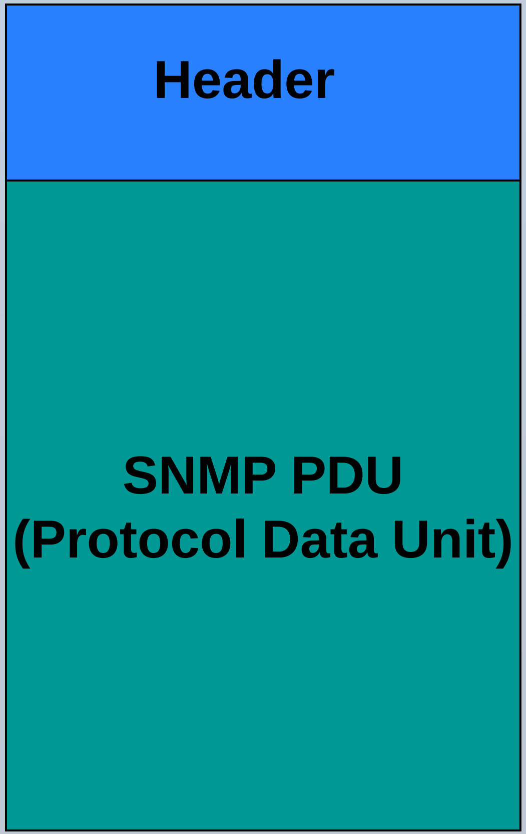
Operation	SNMPv1	SNMPv2c	SNMPv3
Get	Yes	Yes	Yes
GetNext	Yes	Yes	Yes
GetBulk	-	Yes	Yes
Set	Yes	Yes	Yes
Trap	Yes (v1-Trap)	Yes (v2-Trap)	Yes (v2-Trap)
Inform	-	Yes	Yes

SNMP - Messages by Protocol Version

Message	SNMPv1	SNMPv2c	SNMPv3
GetRequest	Yes	Yes	Yes
GetNextRequest	Yes	Yes	Yes
GetBulkRequest	-	Yes	Yes
SetRequest	Yes	Yes	Yes
v1-Trap	Yes	-	-
v2-Trap	-	Yes	Yes
InformRequest	-	Yes	Yes
Response	Yes	Yes	Yes
Report	-	-	Yes

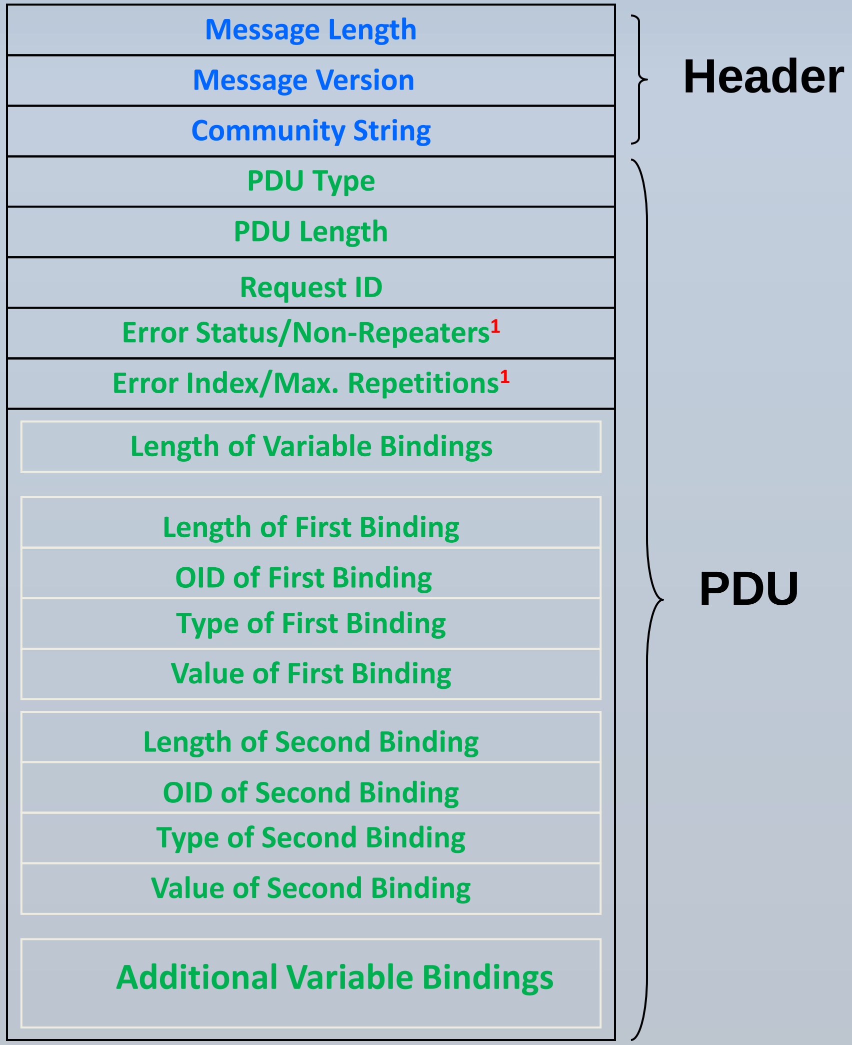
SNMP – Message Format

SNMP message

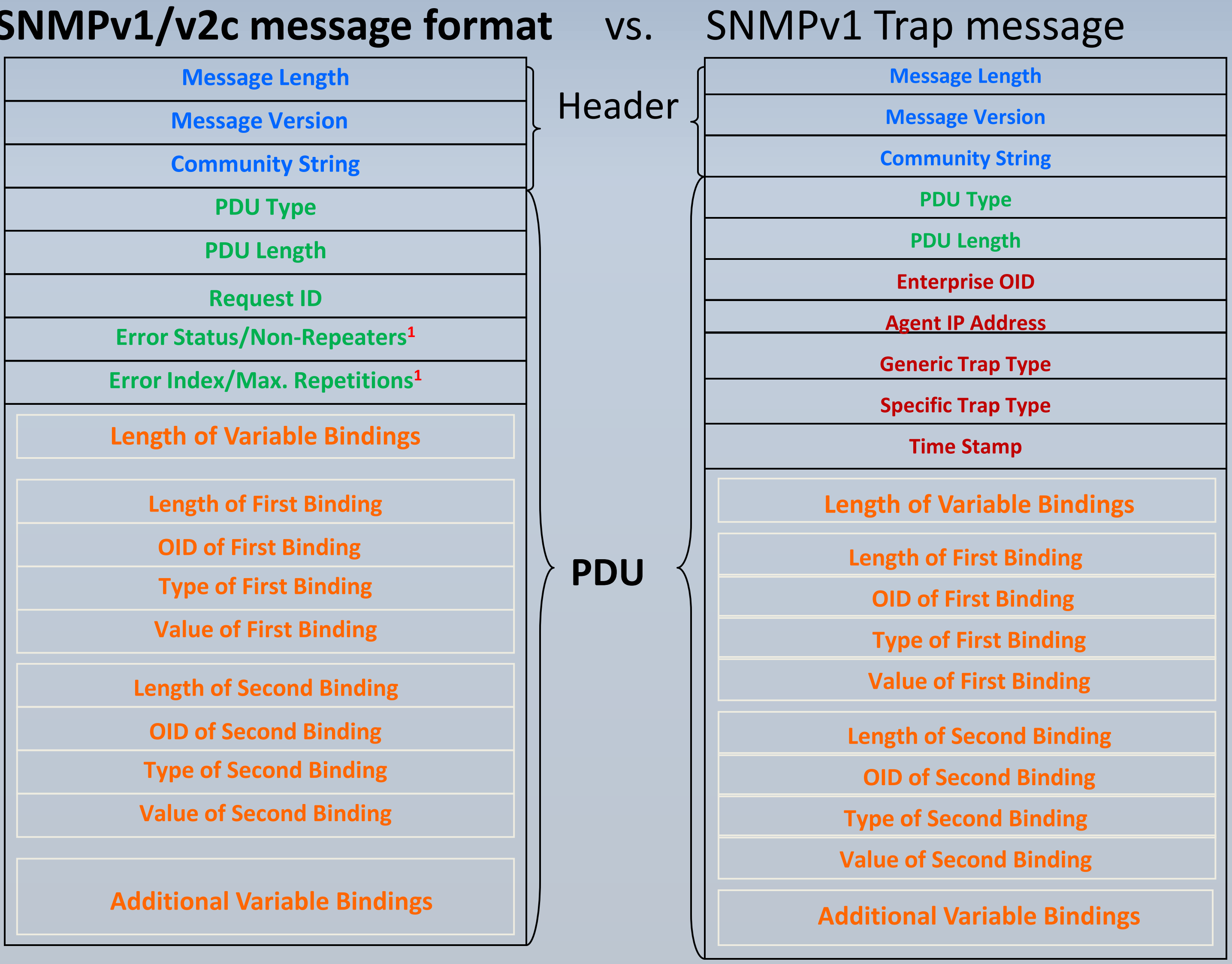


SNMPv1/v2c message format

Get
GetNext
GetBulk[1]
Set
v2-Trap
Inform
Response



SNMP – Message Format



Get
GetNext
GetBulk[1]
Set
v2-Trap
Inform
Response

SNMP – Variables, Variable Bindings

In SNMP:

variable = MIB object instance (=MIB object OID + instance subidentifier)

e.g.: sysUpTime.0 (=1.3.6.1.2.1.1.3.0)

variable binding = **variable** + type + value.

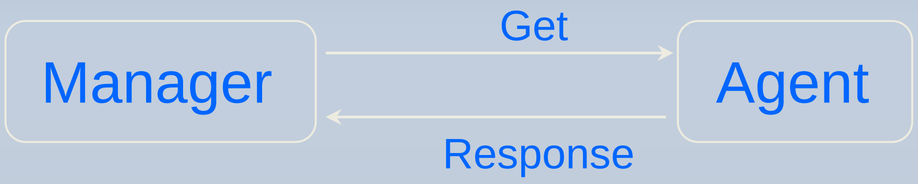
e.g.: sysUpTime.0 (timeticks) 5 days 04h:59m:58s.04th

value of variables in SNMP messages:

In retrieval requests (Get, GetNext, GetBulk), the value field is NULL, a "placeholder".
In a SetRequest or in a Response message carrying requested data, the value portion in variable bindings is present.

SNMP – Operations: Get

1. SNMP Get Operation



SNMP Get operation involves exchange of
two SNMP messages: GetRequest and Response.

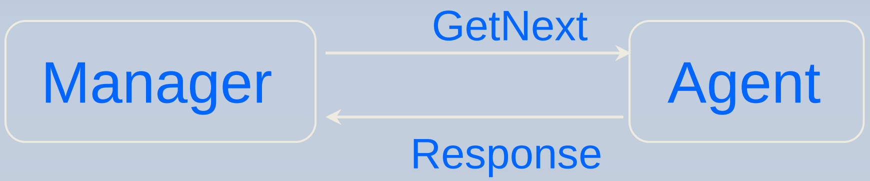
Get operation retrieves the value of one or more variable (MIB object instance).

Get operation requires that the **exact instance of MIB object** whose value is to be retrieved is specified. For example, to retrieve how long the device (agent) has been running since the last reboot:

Get sysUpTime.0 (= Get 1.3.6.1.2.1.1.3.0)
sysUpTime.0 (timeticks) 5 days 04h:59m:58s.04th

SNMP – Operations: GetNext

2. SNMP GetNext Operation



An SNMP GetNext operation involves exchange of
two SNMP messages: GetNextRequest and Response.

GetNext operation retrieves the value of the first object instance that in lexicographical order follows the OID requested in the GetNext PDU.

This operation is used for **dicovering object instances** (and their values) that exist in the queried SNMP agent (e.g., useful for columnar objects). The entire MIB (values of all variables) of an agent can be retrieved by *iterative application of GetNext request* starting at OID 0, each time inserting the last returned OID in the next GetNext request. This operation is called "**SNMP Walk**".

Example: retrieve the description of the first interface in the interfaces table (ifTable):

GetNext ifDescr   (= GetNext 1.3.6.1.2.1.2.2.1.2)
ifDescr.1 (octet string) FastEthernet0/1

SNMP – Operations: GetNext (Cont.)

Example of SNMP Walk using GetNext requests

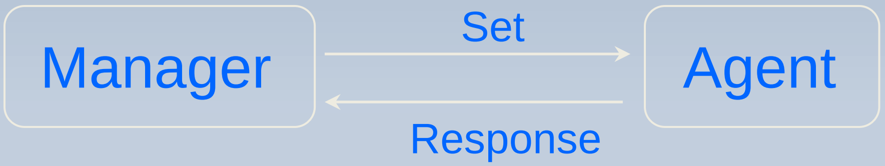
Example: retrieve the infromation from the entire system subtree (1.3.6.1.2.1.1):

GetNext system	(= GetNext 1.3.6.1.2.1.1)
sysDescr.0 (octets) Cisco IOS	(= 1.3.6.1.2.1.1.1.0 (octets) Cisco...)
GetNext sysDescr.0	(= GetNext 1.3.6.1.2.1.1.10)
sysObjectID.0 (oid) catalyst3524XL	(=1.3.6.1.2.1.1.2.0 (oid) catalyst...)
GetNext sysObjectID.0	(= GetNext 1.3.6.1.2.1.1.2.0)
sysUpTime.0 (timeticks) 0 days 13h:31m:14s.11th	(=1.3.6.1.2.1.1.3.0 (timeticks) 0 days...)
GetNext sysUpTime.0	(= GetNext 1.3.6.1.2.1.1.3.0)
sysContact.0 (octets) webmaster@mg-soft.si	(=1.3.6.1.2.1.1.4.0 (octets) webmaster...)
GetNext sysContact.0	(= GetNext 1.3.6.1.2.1.1.4.0)
sysName.0 (octets) Switch1	(=1.3.6.1.2.1.1.5.0 (octets) Switch1)
GetNext sysName.0	(= GetNext 1.3.6.1.2.1.1.5.0)
sysLocation.0 (octets) MG-SOFT Labs, Maribor	(=1.3.6.1.2.1.1.6.0 (octets) MG-SOFT...)
GetNext sysLocation.0	(= GetNext 1.3.6.1.2.1.1.6.0)
sysServices.0 (integer) 2	(=1.3.6.1.2.1.1.7.0 (integer) 2)

....

SNMP – Operations: Set

3. SNMP Set Operation



An SNMP Set operation involves exchange of two SNMP messages:
SetRequest and Response (the latter confirming success or error).

SNMP Set: Used to configure and control managed systems (SNMP devices)

a) **Modifies the value of (one or more) object instance.** Applies to writable objects.

examples:

Set sysContact.0 (octet string) admin@mg-soft.si, Tel:+386-2-2506565 (set system contact info)
sysContact.0 (octet string) admin@mg-soft.si, Tel:+386-2-2506565 (system contact info is set)

Set ifAdminStatus.2 (integer) down(2) (Disable network interface 2)
ifAdminStatus.2 (integer) down(2) (Interface 2 successfully disabled)

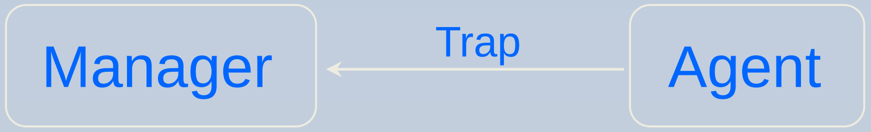
b) In SNMP tables that allow adding/deleting rows via SNMP:

- **Create an instance** of a managed object (adding rows – instantiate a new instance of columnar objects by setting a non-existing instance)
- **Delete an instance** of a managed object (remove rows – remove existing instance (y) of columnar objects by setting the RowStatus.y object to “destroy(6)”)

SNMP – Operations: Trap

Event Reporting Operation

4. SNMP Trap Operation



An SNMP Trap operation involves sending **one** message from agent to manager: SNMP Trap message.

SNMP Trap: Used to notify SNMP manager of an event that occurred on the managed device.

Traps are sent autonomously by SNMP agents when pre-defined events occur, e.g.:

- A network interface on the device (where the agent is running) has gone down.
- Device has been rebooted.
- The fan on a switch or router has failed….

Trap reception is not confirmed (no response is sent back to agent). Unreliable notification.

Different formats of **v1-Trap-PDU** (SNMPv1 Trap message) and **v2-Trap-PDU** (SNMPv2c and SNMPv3 Trap message).

SNMP protocol defines **generic** traps, vendors define **enterprise-specific** traps in their private MIB files.

SNMP – Operations: Trap (Cont.)

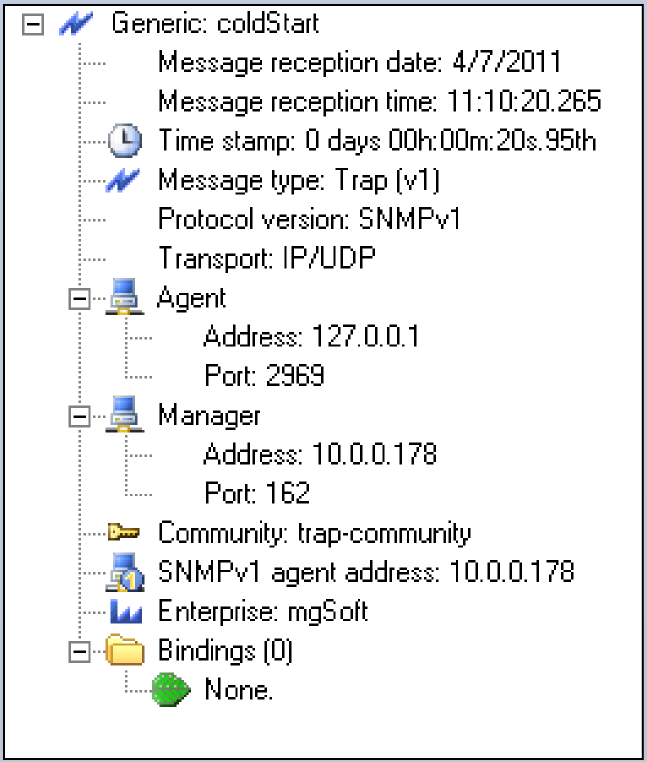
Generic Trap	Meaning
coldStart (0)	Indicates that the agent has rebooted. All management variables will be reset; typically, counters will be reset to zero (0).
warmStart (1)	Indicates that the agent has reinitialized itself. None of the management variables will be reset.
linkDown (2)	Sent when an interface on a device goes down. The first variable binding identifies which interface went down.
linkUp (3)	Sent when an interface on a device comes back up. The first variable binding identifies which interface came back up.
authenticationFailure (4)	Indicates that the agent has been accessed with protocol message that was not properly authenticated (e.g., wrong community string); useful in determining if someone is trying to gain unauthorized access to device.
egpNeighborLoss (5)	Indicates that an Exterior Gateway Protocol (EGP) neighbour has gone down (obsolete).
enterpriseSpecific (6)	Indicates that the trap is enterprise-specific. SNMP vendors and users define their own traps under the private-enterprise branch of the SMI object tree. To process this trap properly, the NMS has to decode the specific trap number that is part of the SNMP message.

SNMP – Operations: Trap (Cont.)

Example: Generic Trap (v1)

```
coldStart TRAP-TYPE
        ENTERPRISE  snmp
        DESCRIPTION
           "A coldStart trap signifies that the sending
           protocol entity is reinitializing itself such
           that the agent's configuration or the protocol
           entity implementation may be altered."
        ::= 0
```

Definition of the SNMPv1 coldStart Trap
(SMIv1 syntax: TRAP-TYPE)

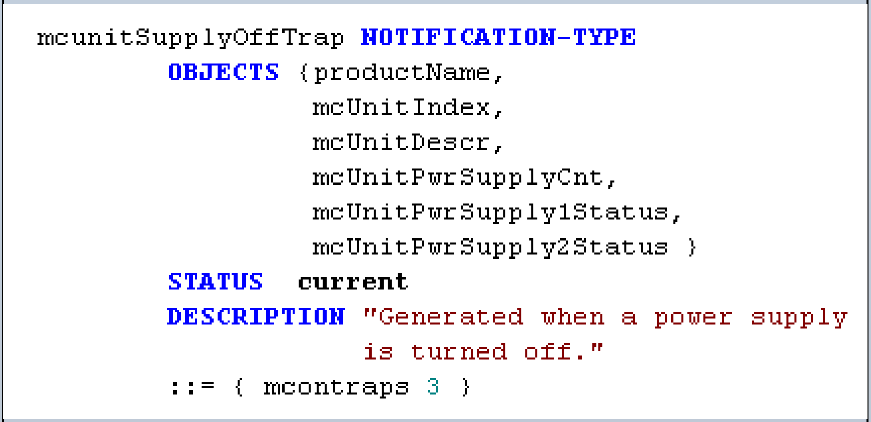


```
Generic: coldStart
    Message reception date: 4/7/2011
    Message reception time: 11:10:20.265
    Time stamp: 0 days 00h:00m:20s.95th
    Message type: Trap (v1)
    Protocol version: SNMPv1
    Transport: IP/UDP
    Agent
        Address: 127.0.0.1
        Port: 2969
    Manager
        Address: 10.0.0.178
        Port: 162
    Community: trap-community
    SNMPv1 agent address: 10.0.0.178
    Enterprise: mgSoft
    Bindings (0)
        None.
```

Details of a SNMPv1 coldStart Trap msg.
displayed by MG-SOFT MIB Browser

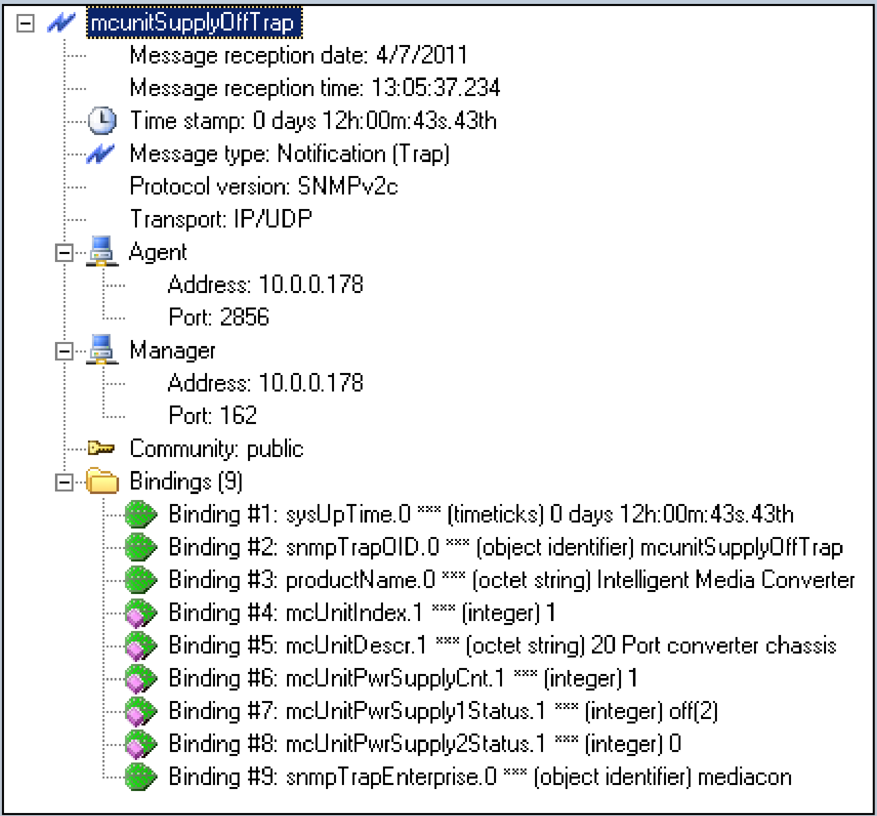
SNMP – Operations: Trap (Cont.)

Example: Enterprise-Specific Trap (v2)



```
mcunitSupplyOffTrap NOTIFICATION-TYPE
        OBJECTS {productName,
                 mcUnitIndex,
                 mcUnitDescr,
                 mcUnitPwrSupplyCnt,
                 mcUnitPwrSupply1Status,
                 mcUnitPwrSupply2Status }
        STATUS   current
        DESCRIPTION "Generated when a power supply
                    is turned off."
        ::= { mcontraps 3 }
```

Definition of a vendor-specific Trap (SMIv2 syntax: NOTIFICATION-TYPE)



```
mcunitSupplyOffTrap
    Message reception date: 4/7/2011
    Message reception time: 13:05:37.234
    Time stamp: 0 days 12h:00m:43s.43th
    Message type: Notification (Trap)
    Protocol version: SNMPv2c
    Transport: IP/UDP
    Agent
        Address: 10.0.0.178
        Port: 2856
    Manager
        Address: 10.0.0.178
        Port: 162
    Community: public
    Bindings (9)
        Binding #1: sysUpTime.0 *** (timeticks) 0 days 12h:00m:43s.43th
        Binding #2: snmpTrapOID.0 *** (object identifier) mcunitSupplyOffTrap
        Binding #3: productName.0 *** (octet string) Intelligent Media Converter
        Binding #4: mcUnitIndex.1 *** (integer) 1
        Binding #5: mcUnitDescr.1 *** (octet string) 20 Port converter chassis
        Binding #6: mcUnitPwrSupplyCnt.1 *** (integer) 1
        Binding #7: mcUnitPwrSupply1Status.1 *** (integer) off(2)
        Binding #8: mcUnitPwrSupply2Status.1 *** (integer) 0
        Binding #9: snmpTrapEnterprise.0 *** (object identifier) mediacon
```

Details of a vendor-specific SNMPv2c Trap displayed by MG-SOFT MIB Browser

SNMP – Operations: Trap (Cont.)

How SNMP managers identify incoming SNMP Trap messages?

SNMP managers need to know what the trap means and how to interpret the information it carries.

SNMPv1-Trap messages:

First, the value of the *generic trap* field in SNMPv1 Trap PDU is checked.

- If the *generic trap* value is **0-5**, the trap is identified as generic of certain type (e.g., coldStart, warmStart, linkDown, linkUp,...).
- If the *generic trap* value is **6**, this is "enterprise-specific“ trap, further identified by the *specific trap* value and *Enterprise OID* value (its name is resolved through the corresponding MIB definition).

SNMPv2-Trap messages (SNMPv2c & SNMPv3 Traps):

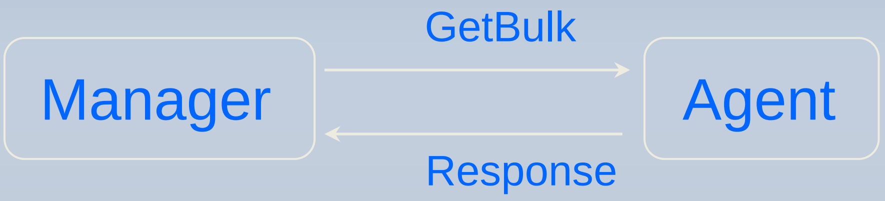
The value of the second variable binding (an OID) identifies a v2-trap message. Same applies to Inform messages.

Note: The bulk of the information about the event that is reported is carried in the variable bindings included in the Trap PDU or Inform PDU!

SNMP – Operations: GetBulk

- Introduced in SNMPv2c -

5. SNMP GetBulk Operation



An SNMP GetBulk operation involves exchange of two SNMP messages: GetBulkRequest and Response.

GetBulk operation is optimization of the GetNext operation. It allows retrieving a large section of a SNMP table (e.g., entire columns or table) with one operation.

It is designed for requesting multiple variables containing columnar and/or scalar objects. The variable bindings list starts with N scalars (N=0,1,2,..), and the remaining M are columnars (M=0,1,2,...). For scalars (if any), one instance is retrieved (regular GetNext is performed), for columnars (if any) more than one instance can be retrieved (more than one GetNext performed internally by agent).

Two additional parameters (special fields in GetBulkRequest PDU):

Non-repeaters: **Number of variable bindings, from the beginning of the MVB list, for which only one instance is requested.**

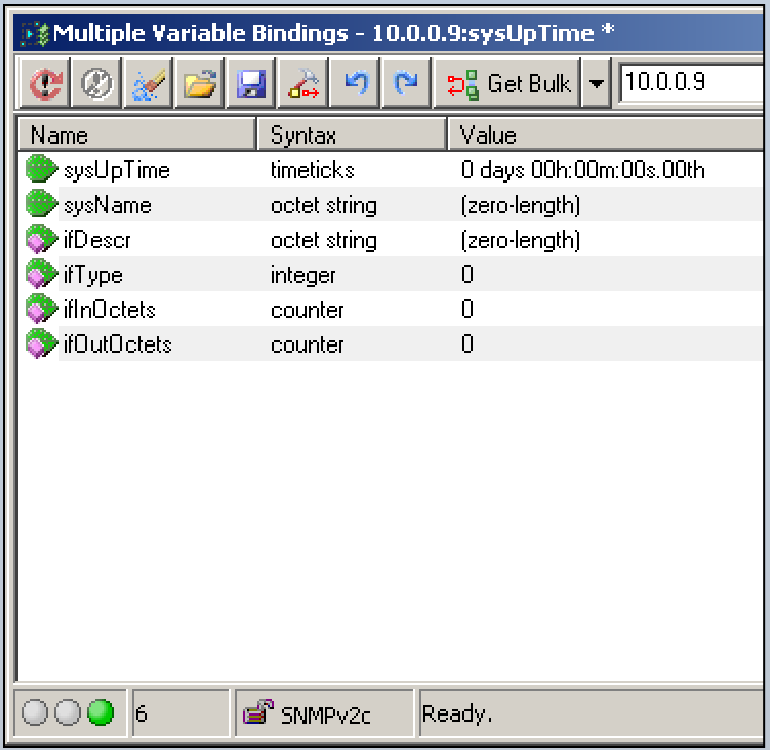
Max. Repetitions: **Max. number of requested instances for the remaining variable bindings (i.e., those listed after non-repeaters).**

SNMP – Operations: GetBulk (Cont.)

GetBulk usage example:

Retrieve 2 scalars (1 instance) and 4 columnars (3 instances each)

1) MVB list **before** performing a GetBulk operation

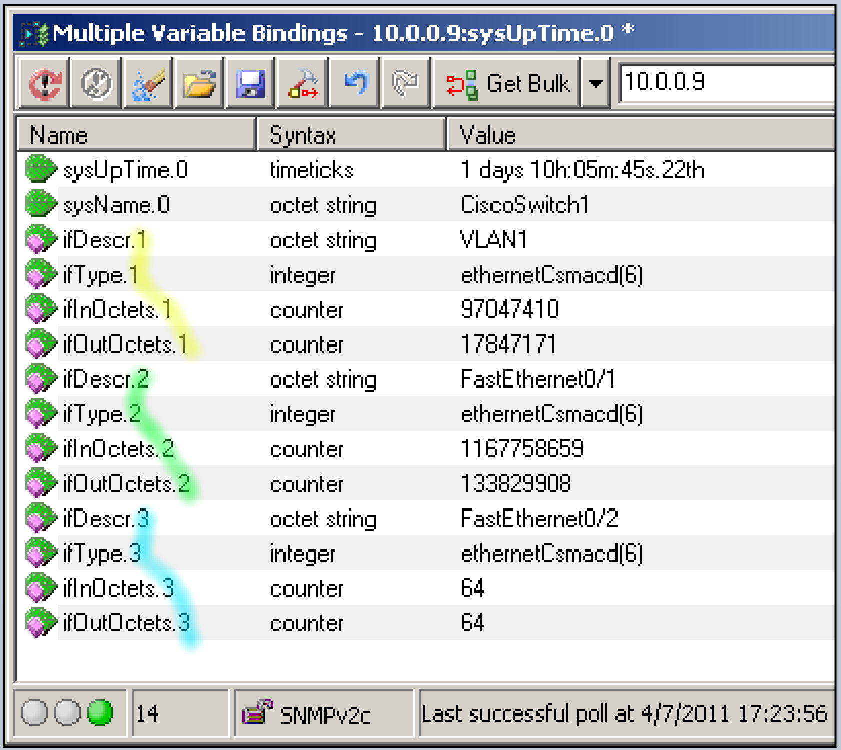


Multiple Variable Bindings - 10.0.0.9:sysUpTime *

Name	Syntax	Value
sysUpTime	timeticks	0 days 00h:00m:00s.00th
sysName	octet string	(zero-length)
ifDescr	octet string	(zero-length)
ifType	integer	0
ifInOctets	counter	0
ifOutOctets	counter	0

6 | SNMPv2c | Ready.

2) MVB list **after** performing a GetBulk operation



Multiple Variable Bindings - 10.0.0.9:sysUpTime.0 *

Name	Syntax	Value
sysUpTime.0	timeticks	1 days 10h:05m:45s.22th
sysName.0	octet string	CiscoSwitch1
ifDescr.1	octet string	VLAN1
ifType.1	integer	ethernetCsmacd(6)
ifInOctets.1	counter	97047410
ifOutOctets.1	counter	17847171
ifDescr.2	octet string	FastEthernet0/1
ifType.2	integer	ethernetCsmacd(6)
ifInOctets.2	counter	1167758659
ifOutOctets.2	counter	133829908
ifDescr.3	octet string	FastEthernet0/2
ifType.3	integer	ethernetCsmacd(6)
ifInOctets.3	counter	64
ifOutOctets.3	counter	64

14 | SNMPv2c | Last successful poll at 4/7/2011 17:23:56

Requested bindings = N+(M*R) => 2+(3*4)=14

N = non-repeaters

M = max-repetitions

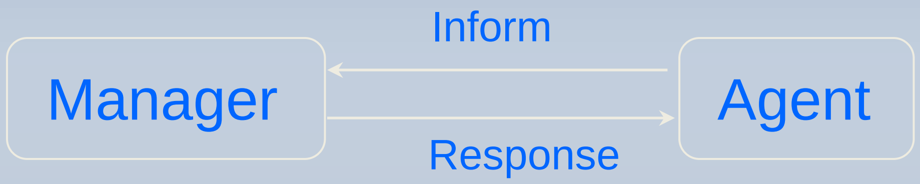
R = number of columnars

(MG-SOFT MIB Browser Pro – MVB window:
14 variable bindings retrieved)

SNMP – Operations: Inform

- Introduced in SNMPv2c -

6. SNMP Inform Operation



An SNMP Inform operation involves exchange of two SNMP messages: InformRequest and Response.

Inform is the “confirmed” event reporting operation. It is used to notify SNMP managers of an event that occurred on the managed device.

Unlike Trap operation that lacks the reception confirmation mechanism, the receiver of an Inform message sends a Response to the sender, acknowledging receipt of the event report (Response message contains the same payload as InfromRequest). If no Response is received before timeout, the InformRequest is retransmitted (same as with other SNMP requests).

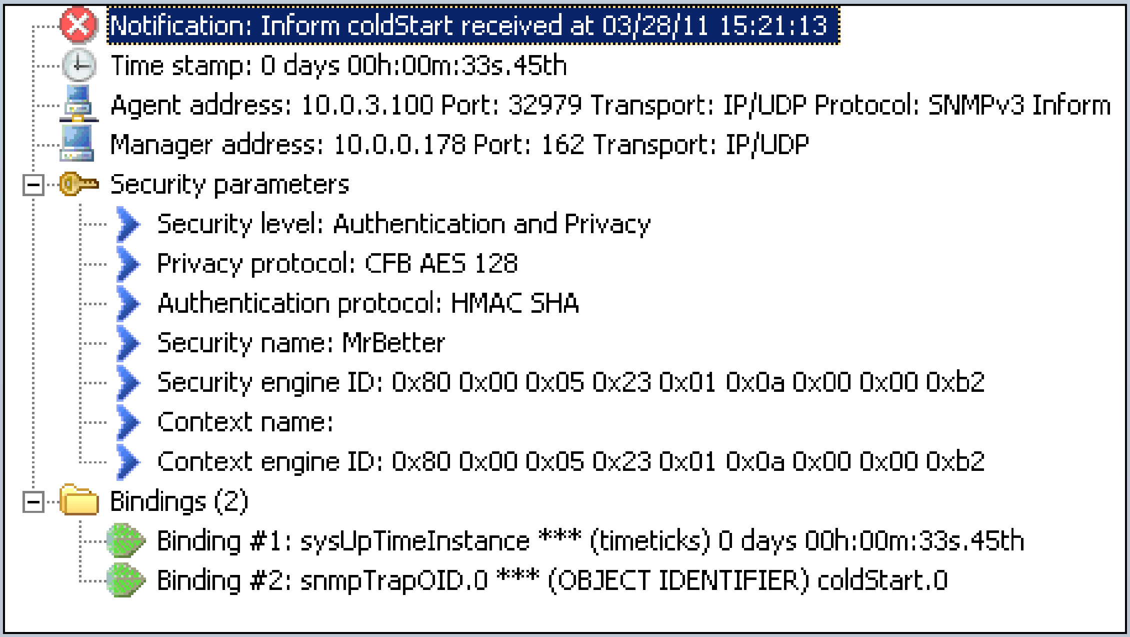
Inform operation was originally intended for manager-to-manager communication, but is now widely used as a means of sending event reports from agent to managers (as alternative to the Trap operation).

SNMP – Operations: Inform (Cont.)

Inform operation is available in SNMPv2c and SNMPv3.

Inform notification messages contain the same payload (variable bindings) as v2-Trap messages and are identified by managers in the same way.

Example: Received SNMPv3 coldStart Inform message (employing SNMPv3 authentication and privacy), as displayed by MG-SOFT Trap Ringer Pro

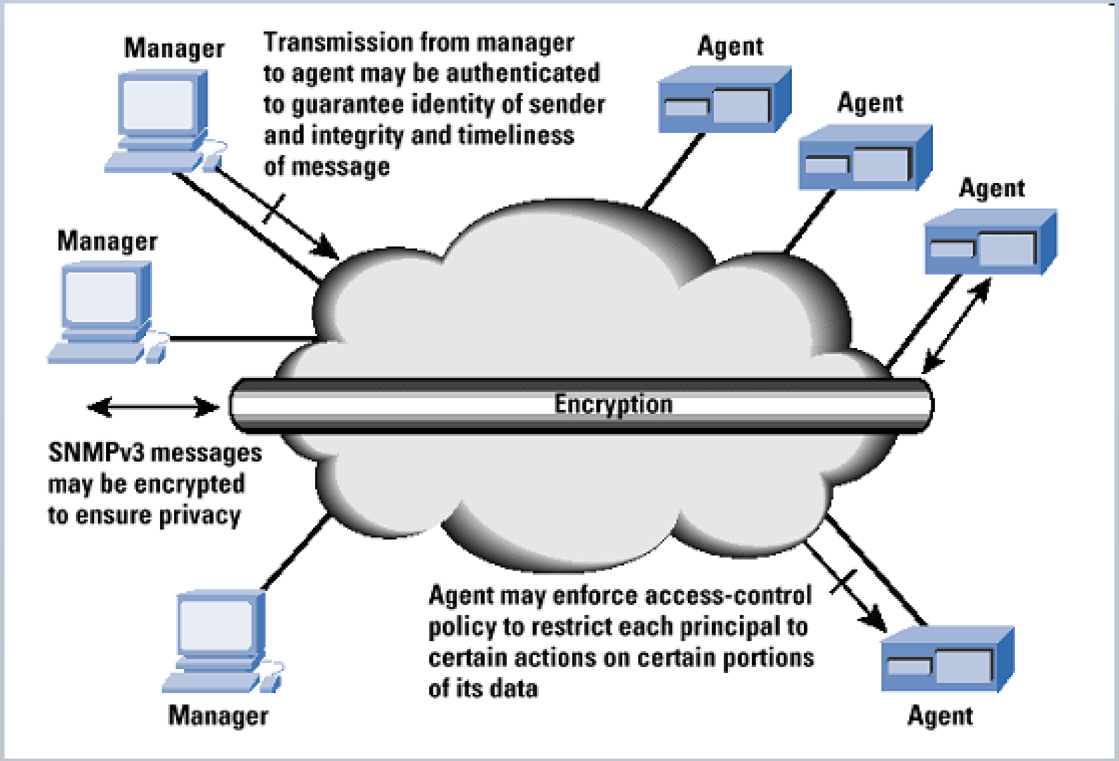


```
Notification: Inform coldStart received at 03/28/11 15:21:13
Time stamp: 0 days 00h:00m:33s.45th
Agent address: 10.0.3.100 Port: 32979 Transport: IP/UDP Protocol: SNMPv3 Inform
Manager address: 10.0.0.178 Port: 162 Transport: IP/UDP
Security parameters
    Security level: Authentication and Privacy
    Privacy protocol: CFB AES 128
    Authentication protocol: HMAC SHA
    Security name: MrBetter
    Security engine ID: 0x80 0x00 0x05 0x23 0x01 0x0a 0x00 0x00 0xb2
    Context name:
    Context engine ID: 0x80 0x00 0x05 0x23 0x01 0x0a 0x00 0x00 0xb2
Bindings (2)
    Binding #1: sysUpTimeInstance *** (timeticks) 0 days 00h:00m:33s.45th
    Binding #2: snmpTrapOID.0 *** (OBJECT IDENTIFIER) coldStart.0
```

SNMPv3 – What it Brings?

SNMPv3 introduces no new protocol operations (it uses SNMPv2c protocol operations and PDUs).

SNMPv3 adds security: authentication, privacy (USM or ISMS), and access control (VACM).


Manager
Transmission from manager to agent may be authenticated to guarantee identity of sender and integrity and timeliness of message
Agent
Agent
Agent
Manager
Encryption
SNMPv3 messages may be encrypted to ensure privacy
Agent may enforce access-control policy to restrict each principal to certain actions on certain portions of its data
Manager
Agent

USM: User-based security model

Operations are perfomed on account of a principal – user (identified by a username) with which security information is associated (auth., privacy, access).

VACM: View-based Acess Control Model

The access control makes it possible to configure agents to provide different levels and scopes of access to the agent's MIB for different users.

New

ISMS: Integrated Security model for SNMP

New alternative to USM. It uses X.509 certificates and TLS/DTLS transport.

SNMPv3 – Security

SNMPv3 USM / VACM security threat protection:

- Masquerade *(verifies the identity of the message's origin by checking the integrity of data)*
- Modification of Information *(thwarts in-transit message alterations by checking the integrity of data)*
- Message Stream Modification *(thwarts replay attack by checking message stream integrity including a time stamp)*
- Disclosure *(prevents eavesdropping by protocol analyzers by using encryption)*
- Unauthorized Access *(Verifies operator authorization by using Access Control Table)*

SNMPv3 – Security (Cont.)

Configuration parameters of USM model:

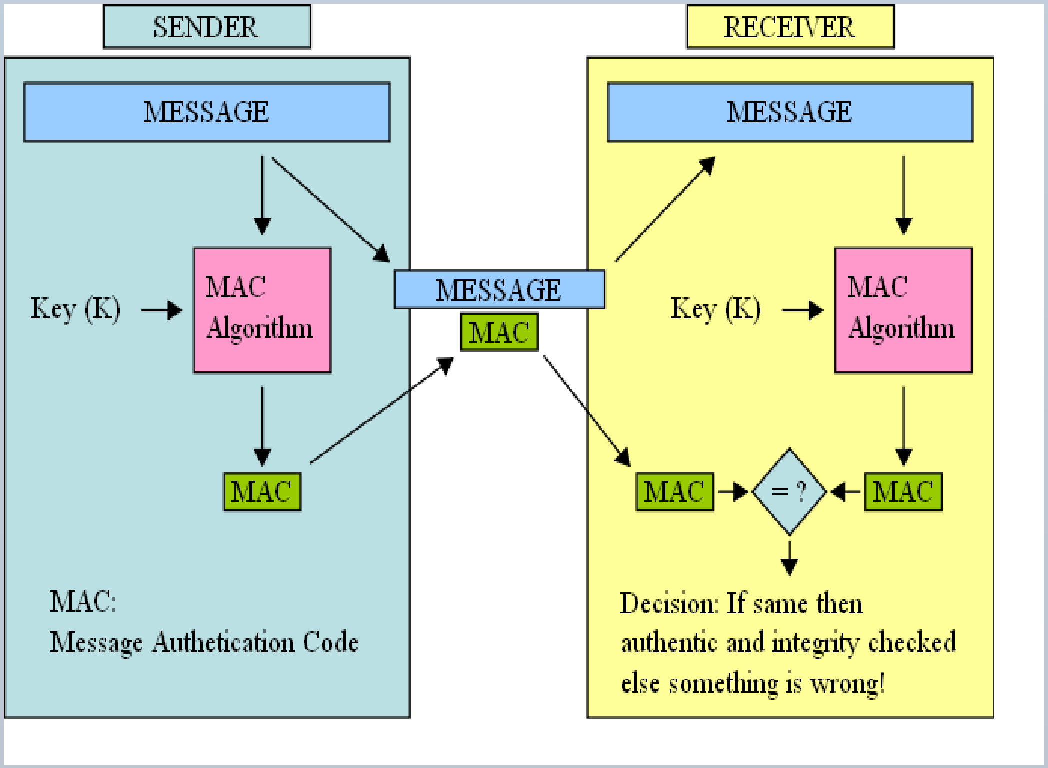
- Security name
- Context name
- Context engine ID
- Authentication protocol
- Privacy protocol
- Authentication password/key
- Privacy password/key

Security levels:

- Without authentication and privacy (noAuthNoPriv)
- Authentication without privacy (authNoPriv)
- Authentication and privacy (authPriv)

SNMPv3 – Security (Cont.)

SNMPv3 USM Message Authentication Mechanism

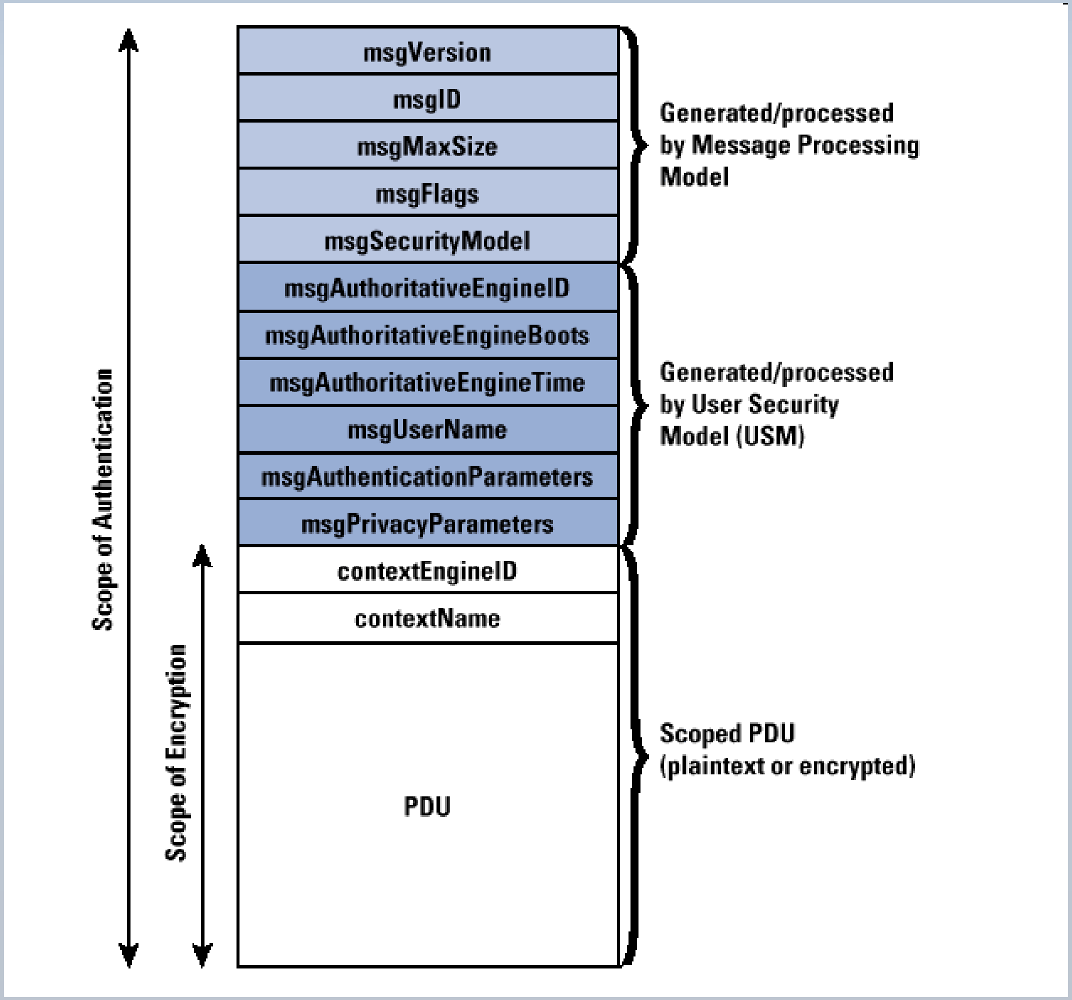

SENDER
RECEIVER
MESSAGE
Key (K)
MAC Algorithm
MAC
MESSAGE
MAC
MAC:
Message Authetication Code
MESSAGE
Key (K)
MAC Algorithm
MAC
= ?
MAC
Decision: If same then authentic and integrity checked else something is wrong!

Authentication protocol provides a mechanism for ensuring SNMPv3 message integrity and origin.

This is achieved by employing HMAC, i.e., passing the message through the one-way hash function (e.g., MD5 or SHA-1) and attaching the hash (a.k.a. message digest, hmac) to the SNMP message.

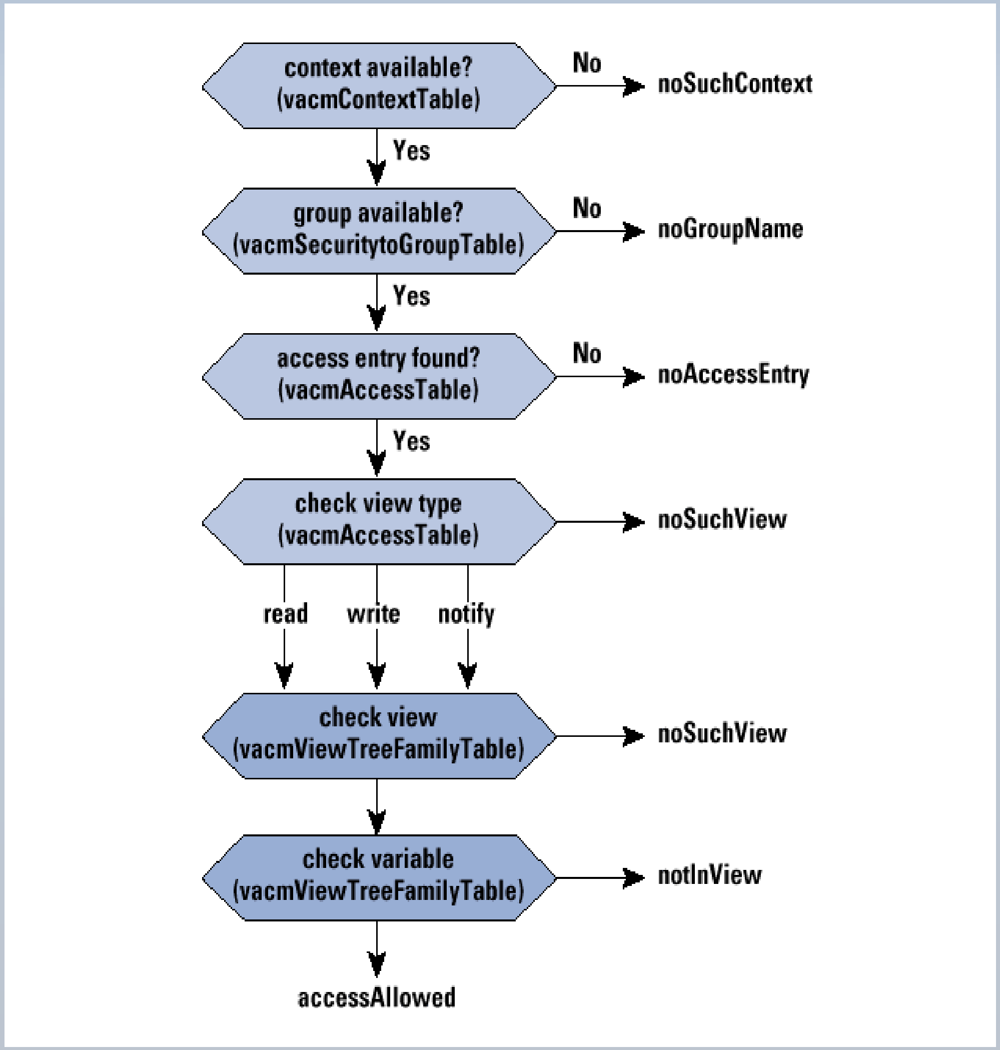
In cryptography, *HMAC* (Hash-based Message Authentication Code) is a specific construction for calculating a message authentication code (MAC) involving a cryptographic hash function in combination with a secret key.

SNMPv3 – Message Format



msgVersion
msgID
msgMaxSize
msgFlags
msgSecurityModel
msgAuthoritativeEngineID
msgAuthoritativeEngineBoots
msgAuthoritativeEngineTime
msgUserName
msgAuthenticationParameters
msgPrivacyParameters
contextEngineID
contextName
PDU
Generated/processed by Message Processing Model
Generated/processed by User Security Model (USM)
Scoped PDU (plaintext or encrypted)
Scope of Authentication
Scope of Encryption

SNMPv3 – VACM



VACM: View-based Acess Control Model
The access control facility makes it possible to configure agents to provide **different levels of access to the agent's MIB to different managers in two ways:**

- **Restrict access to a certain portion of its MIB (MIB View).**
- **Limit the operations that a principal can use on that portion of the MIB (e.g, read-only, read-write, notify)**

Access control policy must be preconfigured in VACM tables on the agent.

Access control is done by group, where a group may be a set of multiple users.

SNMP - Security in General

Security:

- SNMPv1/SNMPv2c
  - ➢Community-based security model
- SNMPv3: User-Based Security Model (USM)
  - ➢Authentication (MD5, SHA-1)
  - ➢ Privacy (DES, AES-128)
  - ➢ Non-standard: Diffie-Hellman key exchange (DOCSIS)
- Integrated Security Model (ISMS) – NEW!
  - ➢ Great improvement in key management
  - ➢ Usage of X.509 certificates, TLS (TCP) and DTLS (UDP) transport, RADIUS

SNMP - Advantages

Advantages of SNMP:

- Simple design
- Widely implemented
- Simple and effective agent implementation (important for embedded systems)
- Extensibility
- Security (SNMPv3)

SNMP - Disadvantages

Disadvantages of SNMP:

- PDU size is not optimized
- Lack of performing operations on managed object
- Data is not well organized
- Active management (SNMP polling wastes bandwidth)
- Security model configuration is not automatic (SNMPv3) or missing (SNMPv1, SNMPv2c).
- Lack of security in SNMPv1 and SNMPv2c resulted in “weak” objects and read-only implementations, that do not support configuring devices effectively via SNMP --> NETCONF was developed.

3. NETCONF

The Network Configuration Protocol (**NETCONF)**, is a network management protocol defined by IETF. Its specification was published in December 2006 as RFC 4741.

NETCONF provides mechanisms to install, manipulate, and delete the configuration (configuration data) of network devices. It also enables monitoring devices (state data).

The NETCONF protocol uses an XML-based data encoding for the configuration data and the protocol messages.

Its operations are realized on top of a simple Remote Procedure Call (RPC) layer.

The default (and mandatory) transport protocol is SSH (secure).

NETCONF (Cont.)

The NETCONF protocol can be conceptually partitioned into four layers:

```
      Layer                          Example
+-------------+    +-------------------------------------------+
|   Content   |    |     Configuration data                    |
+-------------+    +-------------------------------------------+
        |                          |
+-------------+    +-------------------------------------------+
| Operations  |    |<get-config>, <edit-config>, <notification>|
+-------------+    +-------------------------------------------+
        |                          |                    |
+-------------+    +-----------------------------+      |
|     RPC     |    |    <rpc>, <rpc-reply>       |      |
+-------------+    +-----------------------------+      |
        |                          |                    |
+-------------+    +-------------------------------------------+
|  Transport  |    |   BEEP, SSH, SSL, console                 |
|  Protocol   |    |                                           |
+-------------+    +-------------------------------------------+
```

NETCONF (Cont.)

Basic Operations

The base protocol includes the following protocol operations:

- <get>,
- <get-config>,
- <edit-config>,
- <copy-config>,
- <delete-config>,
- <lock>,
- <unlock>,
- <close-session>,
- <kill-session>.

NETCONF (Cont.)

YANG – “human-friendly" modelling language for defining the semantics of operational data, configuration data, notifications, and operations.

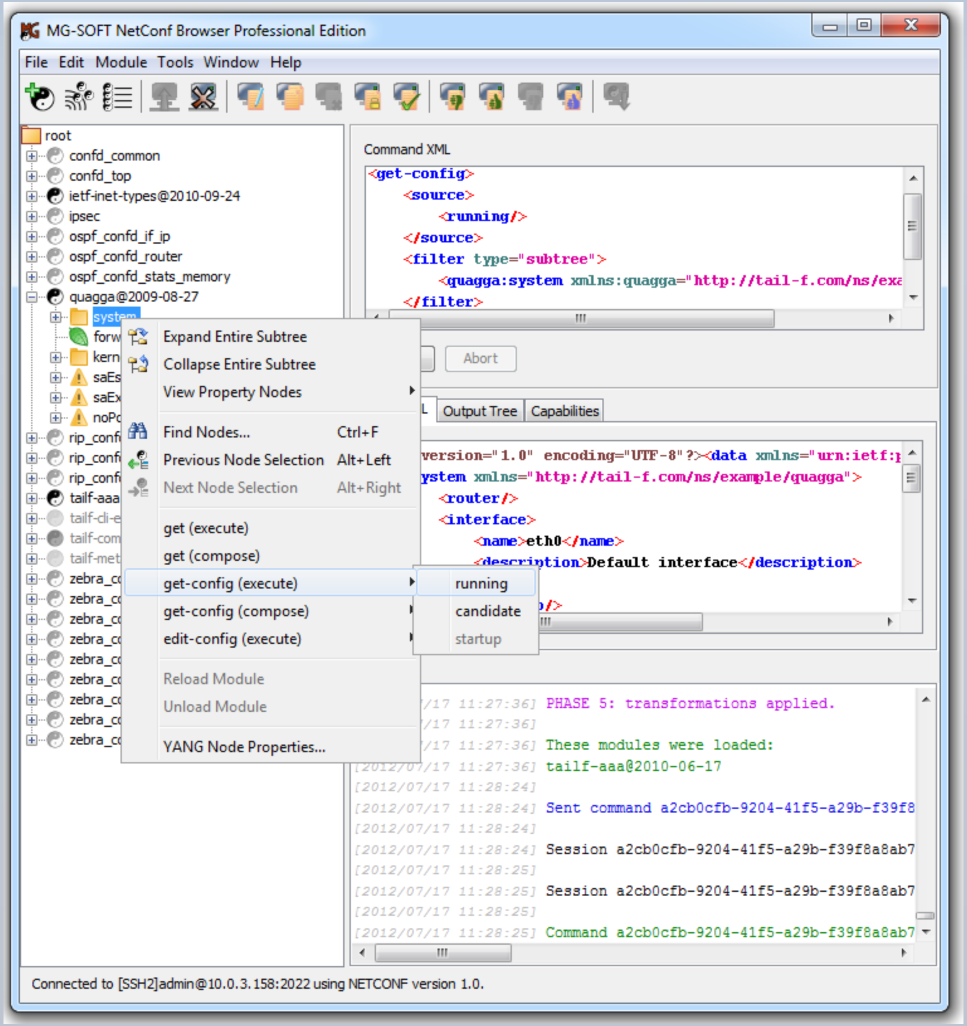
YANG is defined in RFC 6020, and is accompanied by the "Common YANG Data Types" found in RFC 6021.

A collection of related NETCONF management information is written in **YANG modules** that are implemented in NETCONF servers (agents).

YANG models the hierarchical organization of data as a tree in which each node has a name, and either a value or a set of child nodes. YANG provides clear and concise descriptions of the nodes, as well as the interaction between those nodes.

NETCONF (Cont.)

MG-SOFT NETCONF Browser Professional Edition is a powerful and user friendly NETCONF client application that lets you retrieve, modify, install and delete the configuration of any NETCONF server device in the network. The software provides an intuitive GUI that lets you load any valid YANG or YIN module and represent it in form of a hierarchical tree, containing nodes on which NETCONF operations are performed (get, get-config, lock, unlock, edit-config, copy-config, delete-config, etc.).



```
<get-config>
    <source>
        <running/>
    </source>
    <filter type="subtree">
        <quagga:system xmlns:quagga="http://tail-f.com/ns/exa
    </filter>
```

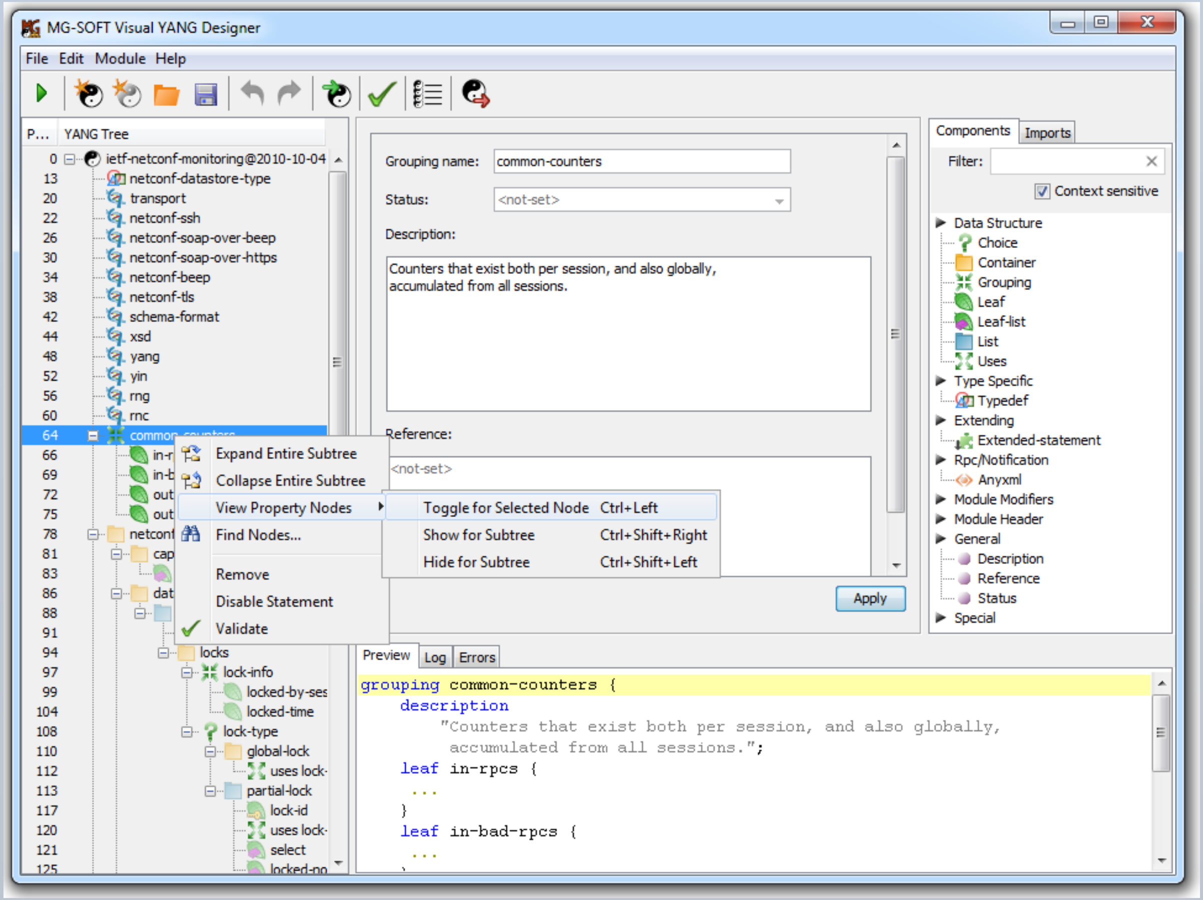
```
version="1.0" encoding="UTF-8"?><data xmlns="urn:ietf:p
ystem xmlns="http://tail-f.com/ns/example/quagga">
  <router/>
  <interface>
    <name>eth0</name>
    <description>Default interface</description>
```

```
/17 11:27:36] PHASE 5: transformations applied.
/17 11:27:36]
/17 11:27:36] These modules were loaded:
[2012/07/17 11:27:36] tailf-aaa@2010-06-17
[2012/07/17 11:28:24]
[2012/07/17 11:28:24] Sent command a2cb0cfb-9204-41f5-a29b-f39f8
[2012/07/17 11:28:24]
[2012/07/17 11:28:24] Session a2cb0cfb-9204-41f5-a29b-f39f8a8ab7
[2012/07/17 11:28:25]
[2012/07/17 11:28:25] Session a2cb0cfb-9204-41f5-a29b-f39f8a8ab7
[2012/07/17 11:28:25]
[2012/07/17 11:28:25] Command a2cb0cfb-9204-41f5-a29b-f39f8a8ab7
```

Connected to [SSH2]admin@10.0.3.158:2022 using NETCONF version 1.0.

NETCONF (Cont.)

MG-SOFT Visual YANG Designer Professional Edition lets you create, design, edit and validate YANG and YIN modules in a visual manner, without having to master the YANG or YIN syntax. The software loads and graphically presents YANG and YIN modules in a tree view. You need to drag & drop YANG objects from the components panel to the tree and assign relevant properties in the application's GUI. Then you only need to save the modules to consistent YANG or YIN files.



Grouping name: common-counters

Status: <not-set>

Description:

Counters that exist both per session, and also globally, accumulated from all sessions.

Reference:

<not-set>

```
grouping common-counters {
    description
        "Counters that exist both per session, and also globally,
         accumulated from all sessions.";
    leaf in-rpcs {
      ...
    }
    leaf in-bad-rpcs {
      ...
```

Contacting MG-SOFT

MG-SOFT d.o.o.
Strma ulica 8
SI-2000 Maribor
Slovenia

+386 2 2506565
+386 2 2506566
info@mg-soft.si
http://www.mg-soft.si/