OVERVIEW

BACKGROUND
• HISTORY, GOALS & STANDARDS

• STRUCTURE OF MANAGEMENT INFORMATION
  • SCALARS
  • TABLES

• MANAGEMENT INFORMATION BASES
  • OVERVIEW
    • MIB-II
    • SNMPv2, IF & IP MIB

• SIMPLE NETWORK MANAGEMENT PROTOCOL
  • VERSION 1
  • VERSION 2
  • VERSION 3

AGENTX

DISMAN

FURTHER INFORMATION
SNMP HISTORY

- SGMP
- SNMP
- SNMP security
- SMP
- SNMPv2 (parties)
- SNMPv2 (community)
- SNMPv3

- CMOT

- HEMS/HEMP

- 1987
- 1988
- 1989
- 1990
- 1991
- 1992
- 1993
- 1994
- 1995
- 1996
- 1997
- 1998
- 1999
SNMP GOALS

UBIQUITY
• PCs AND CRAYs

INCLUSION OF MANAGEMENT SHOULD BE INEXPENSIVE
• SMALL CODE
• LIMITED FUNCTIONALITY

MANAGEMENT EXTENSIONS SHOULD BE POSSIBLE
• NEW MIBs

MANAGEMENT SHOULD BE ROBUST
• CONNECTIONLESS TRANSPORT
PRINCIPLE OPERATION

MANAGER

SNMP

AGENTS

MIB
PRINCIPLE OPERATION

MANAGER

AGENTS

POLLING

TRAPS

MIB
PRINCIPLE OPERATION

MANAGER

GET / SET

TRAP

AGENTS
PRINCIPLE OPERATION

MANAGER

AGENTS

TABLES

VARIABLES
SNMP STRUCTURE

MANAGER

AGENT

Management Application

SNMP PDUs

CONNECTIONLESS TRANSPORT SERVICE PROVIDER

UDP

MIB
STANDARDS

SMI
• STRUCTURE OF MANAGEMENT INFORMATION
  • RFC 1155

MIB-II
• MANAGEMENT INFORMATION BASE
  • RFC 1213
• A LARGE NUMBER OF ADDITIONAL MIBs EXIST

SNMP
• SIMPLE NETWORK MANAGEMENT PROTOCOL
  • RFC 1157
• NAME IS USED IN A MORE GENERAL SENSE

NEWER VERSIONS: SNMPv2 & SNMPv3
SMI

STRUCTURE OF MANAGEMENT INFORMATION

RFC 1155: SMIv1
RFC 1212: CONCISE MIB DEFINITIONS
RFC 2578: SMIv2
RFC 2579: TEXTUAL CONVENTIONS

MAKES THE DEFINITION OF (NEW) MIBs EASIER
SMI

MANAGEMENT INFORMATION WITHIN MANAGED SYSTEMS MUST BE REPRESENTED AS:
- SCALARS
- TABLES
(= TWO DIMENSIONAL ARRAYS OF SCALARS)

THE SNMP PROTOCOL CAN ONLY EXCHANGE (A LIST OF) SCALARS

DEFINED IN TERMS OF ASN.1 CONSTRUCTS
### SMI: DATA TYPES FOR SCALARS

<table>
<thead>
<tr>
<th>SIMPLE TYPES:</th>
<th>SMIv1</th>
<th>SMIv2</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTEGER</td>
<td></td>
<td>INTEGER</td>
</tr>
<tr>
<td>OCTET STRING</td>
<td></td>
<td>OCTET STRING</td>
</tr>
<tr>
<td>OBJECT IDENTIFIER</td>
<td></td>
<td>OBJECT IDENTIFIER</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>APPLICATION-WIDE TYPES:</th>
<th>SMIv1</th>
<th>SMIv2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gauge</td>
<td></td>
<td>Unsigned32</td>
</tr>
<tr>
<td>Counter</td>
<td></td>
<td>Gauge32</td>
</tr>
<tr>
<td>TimeTicks</td>
<td></td>
<td>Counter32</td>
</tr>
<tr>
<td>IpAddress</td>
<td></td>
<td>Counter64</td>
</tr>
<tr>
<td>Opaque</td>
<td></td>
<td>TimeTicks</td>
</tr>
<tr>
<td>NetworkAddress</td>
<td></td>
<td>IpAddress</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PSEUDO TYPES:</th>
<th>SMIv1</th>
<th>SMIv2</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td></td>
<td>BITS</td>
</tr>
</tbody>
</table>
EXAMPLE OF SCALAR OBJECTS

MANAGER

AGENT

SNMP

MANAGED OBJECT INSTANCES

name

address

uptime
OBJECT NAMING

INTRODUCE NAMING TREE

NEW-MIB:

1

- address (1)
  - 130.89.16.2
- info (2)
- name (1)
  - printer-1
- uptime (2)
  - 123456

THE LEAVES OF THE TREE REPRESENT THE MANAGED OBJECTS

NODES ARE INTRODUCED FOR NAMING PURPOSES
OBJECT NAMING

- **address**
  Object ID = 1.1
  Object Instance = 1.1.0
  Value of Instance = 130.89.16.2

- **info**
  Object ID = 1.2

- **name**
  Object ID = 1.2.1
  Object Instance = 1.2.1.0
  Value of Instance = *printer-1*

- **uptime**
  Object ID = 1.2.2
  Object Instance = 1.2.2.0
  Value of Instance = 123456

**ALTERNATIVE:**
Object ID = NEW-MIB  info  uptime
OBJECT NAMING: MIBs

root
  └── ccitt (0)
      ├── iso (1)
      │    └── joint-iso-ccitt (2)
      │         ├── stnd (0)
      │         │    └── reg-auth (1)
      │         │         └── mb (2)
      │         └── org (3)
      │             └── dod (6)
      │                 └── internet (1)
      │                         └── mib-2 (1)
      │                             └── mngt (2)
      │                                 └── experimental (3)
      │                                      └── private (4)
      │                                           └── security (5)
      │                                               └── snmpV2 (6)
      │                                                   └── snmpDomains (1)
      │                                                       └── snmpProxys (2)
      │                                                                 └── snmpModules (3)
OBJECT TYPE DEFINITION

OBJECT-TYPE:

SYNTAX

- INTEGER
- OCTET STRING
- OBJECT IDENTIFIER
- BITS
- IpAddress
- Integer32
- Counter32
- Counter64
- Gauge32
- TimeTicks
- Opaque
- New Type

MAX-ACCESS

- read-only
- read-write
- read-create
- accessible-for-notify
- not-accessible

STATUS

- current
- deprecated
- obsolete

DESCRIPTION

""
TABLES

EXAMPLE: ROUTING TABLE

<table>
<thead>
<tr>
<th>destination</th>
<th>next</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
</tr>
</tbody>
</table>

TO RETRIEVE INDIVIDUAL TABLE ENTRIES

EACH ENTRY SHOULD GET A NAME
POSSIBILITY 1 (NOT BEING USED BY SNMP): USE ROW NUMBERS

NEW-MIB:

1

address (1)
130.89.16.2

info (2)

name (1) printer-1
uptime (2) 123456

routeTable (3)

dest(1)
next(2)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
</tr>
</tbody>
</table>

this is row 5

EXAMPLE: THE VALUE OF NEW-MIB routeTable next 5 IS 3
POSSIBILITY 2 (USED BY SNMP): INTRODUCE AN INDEX COLUMN

NEW-MIB:

EXAMPLE: THE VALUE OF NEW-MIB routeTable next 5 IS 2
TABLE INDEXING

GENERAL SCHEME

X.C.I

OID of Table  Column number  Index value

EXAMPLES:

OID of Table = 1.3
1.3.1.5 => 5
1.3.2.5 => 2
1.3.1.9 => 9
1.3.2.9 => 3
1.3.2.7 => 2
1.3.1.1 => *entry does not exist*
1.3.2.1 => *entry does not exist*
TABLE INDEXING - NON-INTEGER INDEX

AN INDEX NEED NOT BE AN INTEGER

routeTable (3)

dest (1) next (2)

| 130.89.16.1 | 130.89.16.1 |
| 130.89.16.4 | 130.89.16.4 |
| 130.89.16.23 | 130.89.16.1 |
| 130.89.19.121 | 130.89.16.1 |
| 192.1.23.24 | 130.89.16.4 |
| 193.22.11.97 | 130.89.16.4 |

EXAMPLES:
OID of Table = 1.3
1.3.1.130.89.16.23 => 130.89.16.23
1.3.2.130.89.16.23 => 130.89.16.1
1.3.1.193.22.11.97 => 193.22.11.97
1.3.2.193.22.11.97 => 130.89.16.4
1.3.2.130.89.19.121 => 130.89.16.1
TABLE INDEXING - MULTIPLE INDEX FIELDS

USE OF MULTIPLE INDEX FIELDS

X.C.I1.I2

- OID of Table
- Column number
- Index value 1
- Index value 2
**EXAMPLE:**

1.3.3.192.1.23.24.1 \(\Rightarrow\) 130.89.16.1

1.3.3.192.1.23.24.2 \(\Rightarrow\) 130.89.16.4
DEFINITION OF NEW TYPES

TEXTUAL CONVENTIONS

TO REFINE SEMANTICS OF EXISTING TYPES

EXAMPLE:

RunState ::= TEXTUAL CONVENTION
STATUS current
DESCRIPTION "..."
SYNTAX INTEGER{
    running(1)
    runnable(2)
    waiting(3)
    exiting(4)
TEXTUAL CONVENTIONS

- PhysAddress
- MacAddress
- TruthValue
- AutonomousType
- InstancePointer
- VariablePointer
- RowPointer
- RowStatus
- TimeStamp
- TimeInterval
- DateAndTime
- StorageType
- TDomain
- TAddress
- Inet-Address...
SMIv2:
• MIBs MAY NOW INCLUDE NOTIFICATION TYPE MACROS

EXAMPLE:
linkUp  NOTIFICATION-TYPE
OBJECTS   {ifIndex}
STATUS    current
DESCRIPTION
"A linkUp trap signifies that the entity has detected that the ifOperStatus object has changed to Up"
::= {snmpTraps 4}
SMIng

PROBLEMS WITH SMIv2

• SMIv2 RELIED ON 1988 VERSION OF ASN.1

• TOOLS FOR SMIv2 RELATIVELY COMPLEX

• CERTAIN DATA TYPES WERE MISSING IN SMIv2
  64 bit integers, ...

• LIMITED FACILITIES TO REUSE DEFINITIONS

• SMIv2 DID NOT ALLOW FOR EXTENSIONS

• NEW, POSSIBLY INCOMPATIBLE VARIANTS APPEARED
  SPPI, ...
SMIng

TO RESOLVE THESE PROBLEMS
A NEW SMI IS BEING DEFINED

SMI next generation (ng)

STARTED AS IRTF NMRG ACTIVITY

IS NOW IETF WG

THE CHALLENGE:
CREATE A COMMON DATA DEFINITION LANGUAGE,
INDEPENDENT OF SPECIFIC PROTOCOLS
DATA VERSUS INFORMATION MODEL

INFORMATION MODEL:

"formal" / UML, ...

DATA MODEL:

MIBs

PIBs

SMIV2

SPPI

LDAP SCHEMA

CORBA IDL
DATA MODEL & "TRANSFER" PROTOCOL

DATA MODEL:

MIBs

PIBs

"TRANSFER" PROTOCOL:

SNMP - BER

COPS - BER

LDAPv3

CORBA IDL

LDAPv3 SCHEMA

SMIv2

SPPI

COPS - BER
ROLE OF SMIng

INFORMATION MODEL

DATA MODEL:

MIBs / PIBs

SNMP - BER

COPS - BER

LDAP SCHEMA

CORBA IDL
MIBs

MANAGEMENT INFORMATION BASES

CONTAIN THE MANAGED OBJECTS (VARIABLES) THAT REPRESENT THE RESOURCES OF A SYSTEM AND WHICH MAY BE MONITORED AND MODIFIED BY A (REMOTE) MANAGER TO CONTROL THE BEHAVIOUR OF THAT SYSTEM

MANAGER

AGENT

SNMP
MIB DEFINITION AND MIB INSTANCE

MIB DEFINITIONS SHOULD BE KNOWN BY:

• THE IMPLEMENTORS OF THE MANAGED SYSTEM
  • THE MANAGER

THE MIB IS INSTANTIATED WITHIN THE MANAGED SYSTEM
MODULARITY

THE MANAGED OBJECTS OF A SYSTEM ARE USUALLY DEFINED IN MULTIPLE MIB DEFINITIONS

MODULES

• DIFFERENT MODULES CAN BE DEFINED BY DIFFERENT TEAMS

• MANAGEMENT FUNCTIONALITY CAN GRADUALLY BE EXTENDED

• DIFFERENT TYPES OF SYSTEMS CAN SUPPORT DIFFERENT MIB MODULES

• VENDORS CAN EXTEND THE MANAGEMENT FUNCTIONALITY VIA PROPRIETARY MIBS
HARDWARE MIBS

HOST RESOURCES MIB

MODEM MIB

PRINTER MIB
PROTOCOL MIBS

APPLICATION
WWW  MAIL  DNS  X.500  RDBMS  SNMP  ...

TRANSPORT
TCP  UDP

NETWORK
OSPF  IP  ICMP  BGP  ARP  EGP  ...

TRANSMISSION
INTERFACES
802.3  802.5  FDDI  ATM  ADSL  SONET  ...
PROTOCOL MIBS - EXAMPLE: MIB-II

APPLICATION
WWW, MAIL, DNS, X.500, RDBMS, SNMP, ...

TRANSPORT
TCP, UDP

NETWORK
OSPF, BGP, EGP, IP, ICMP, ARP, ...

TRANSMISSION
802.3, 802.5, FDDI, ATM, ADSL, SONET, ...

INTERFACES
# HARDWARE SPECIFIC MIBs

<table>
<thead>
<tr>
<th>Title</th>
<th>RFC</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Resources MIB</td>
<td>2790</td>
<td>D</td>
</tr>
<tr>
<td>Entity MIB</td>
<td>2737</td>
<td>P</td>
</tr>
<tr>
<td>Job Monitoring MIB</td>
<td>2707</td>
<td>I</td>
</tr>
<tr>
<td>Printer</td>
<td>1759</td>
<td>P</td>
</tr>
<tr>
<td>Modem</td>
<td>1696</td>
<td>P</td>
</tr>
<tr>
<td>Parallel printer-like Hardware</td>
<td>1660</td>
<td>D</td>
</tr>
<tr>
<td>RS-232-like Hardware</td>
<td>1659</td>
<td>D</td>
</tr>
<tr>
<td>Character Stream Devices</td>
<td>1658</td>
<td>D</td>
</tr>
<tr>
<td>UPS</td>
<td>1628</td>
<td>P</td>
</tr>
</tbody>
</table>
## TRANSMISSION MIBs

<table>
<thead>
<tr>
<th>Title</th>
<th>RFC</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNI/NNI Multilink Frame Relay function</td>
<td>3020</td>
<td>P</td>
</tr>
<tr>
<td>Frame Relay/ATM PVC Service Interworking Function</td>
<td>2955</td>
<td>P</td>
</tr>
<tr>
<td>Frame Relay Service</td>
<td>2954</td>
<td>P</td>
</tr>
<tr>
<td>Inverted Stack Table Extension to the Interfaces Group</td>
<td>2864</td>
<td>P</td>
</tr>
<tr>
<td>Interfaces Group</td>
<td>2863</td>
<td>D</td>
</tr>
<tr>
<td>Fabric Element in Fibre Channel Standard</td>
<td>2837</td>
<td>P</td>
</tr>
<tr>
<td>NBMA Next Hop Resolution Protocol (NHRP)</td>
<td>2677</td>
<td>P</td>
</tr>
<tr>
<td>Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions</td>
<td>2674</td>
<td>P</td>
</tr>
<tr>
<td>Radio Frequency MIB for MCNS/DOCSIS compliant RF interfaces</td>
<td>2670</td>
<td>P</td>
</tr>
<tr>
<td>Cable Device MIB for DOCSIS compliant Cable Modems and Cable Modem Termination Systems</td>
<td>2669</td>
<td>P</td>
</tr>
<tr>
<td>IEEE 802.3 Medium Attachment Units (MAUs)</td>
<td>2668</td>
<td>P</td>
</tr>
<tr>
<td>Object Identifiers for Identifying Ethernet Chip Sets</td>
<td>2666</td>
<td>I</td>
</tr>
<tr>
<td>Title</td>
<td>RFC</td>
<td>STATUS</td>
</tr>
<tr>
<td>--------------------------------------------------------------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>Ethernet-like Interface Types</td>
<td>2665</td>
<td>P</td>
</tr>
<tr>
<td>ADSL Lines</td>
<td>2662</td>
<td>P</td>
</tr>
<tr>
<td>SONET/SDH Interface Type</td>
<td>2558</td>
<td>P</td>
</tr>
<tr>
<td>ATM Management</td>
<td>2515</td>
<td>P</td>
</tr>
<tr>
<td>Textual Conventions and OBJECT-IDENTITIES for ATM Management</td>
<td>2514</td>
<td>P</td>
</tr>
<tr>
<td>DS3/E3 Interface Type</td>
<td>2496</td>
<td>P</td>
</tr>
<tr>
<td>DS1, E1, DS2 and E2 Interface Types</td>
<td>2495</td>
<td>P</td>
</tr>
<tr>
<td>DS0 and DS0 Bundle Interface Type</td>
<td>2494</td>
<td>P</td>
</tr>
<tr>
<td>Classical IP and ARP Over ATM (IPOA)</td>
<td>2320</td>
<td>P</td>
</tr>
<tr>
<td>IEEE 802.12 Repeater Devices</td>
<td>2266</td>
<td>P</td>
</tr>
<tr>
<td>Dial Control</td>
<td>2128</td>
<td>P</td>
</tr>
<tr>
<td>ISDN</td>
<td>2127</td>
<td>P</td>
</tr>
<tr>
<td>Frame Relay DTEs</td>
<td>2115</td>
<td>D</td>
</tr>
<tr>
<td>IEEE 802.3 Repeater Devices</td>
<td>2108</td>
<td>P</td>
</tr>
<tr>
<td>Data Link Switching</td>
<td>2024</td>
<td>P</td>
</tr>
<tr>
<td>IEEE 802.12 Interfaces</td>
<td>2020</td>
<td>P</td>
</tr>
<tr>
<td>IEEE 802.5 Station Source Routing</td>
<td>1749</td>
<td>P</td>
</tr>
<tr>
<td>Title</td>
<td>RFC</td>
<td>STATUS</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>IEEE 802.5</td>
<td>1748</td>
<td>D</td>
</tr>
<tr>
<td>SMDS</td>
<td>1694</td>
<td>D</td>
</tr>
<tr>
<td>Source Routing Bridges</td>
<td>1525</td>
<td>P</td>
</tr>
<tr>
<td>FDDI</td>
<td>1512</td>
<td>P</td>
</tr>
<tr>
<td>Bridges</td>
<td>1493</td>
<td>D</td>
</tr>
<tr>
<td>Bridge Network Control Protocol of PPP</td>
<td>1474</td>
<td>P</td>
</tr>
<tr>
<td>IP Network Control Protocol of PPP</td>
<td>1473</td>
<td>P</td>
</tr>
<tr>
<td>Security Protocols of PPP</td>
<td>1472</td>
<td>P</td>
</tr>
<tr>
<td>Link Control Protocol of PPP</td>
<td>1471</td>
<td>P</td>
</tr>
<tr>
<td>Multiprotocol Interconnect over X.25</td>
<td>1461</td>
<td>P</td>
</tr>
<tr>
<td>X.25 Packet Layer</td>
<td>1382</td>
<td>P</td>
</tr>
<tr>
<td>X.25 LAPB</td>
<td>1381</td>
<td>P</td>
</tr>
</tbody>
</table>
# NETWORK LAYER MIBs

<table>
<thead>
<tr>
<th>Title</th>
<th>RFC</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv6 MIB for The Multicast Listener Discovery Protocol</td>
<td>3019</td>
<td>P</td>
</tr>
<tr>
<td>Protocol Independent Multicast MIB for IPv4</td>
<td>2934</td>
<td>E</td>
</tr>
<tr>
<td>Internet Group Management Protocol MIB</td>
<td>2933</td>
<td>P</td>
</tr>
<tr>
<td>IPv4 Multicast Routing MIB</td>
<td>2932</td>
<td>P</td>
</tr>
<tr>
<td>Textual Conventions for Internet Network Addresses</td>
<td>2851</td>
<td>P</td>
</tr>
<tr>
<td>Definitions of MO for the Virtual Router Redundancy Protocol</td>
<td>2787</td>
<td>P</td>
</tr>
<tr>
<td>IP Tunnel MIB</td>
<td>2667</td>
<td>P</td>
</tr>
<tr>
<td>MIB for IPv6: ICMPv6 Group</td>
<td>2466</td>
<td>P</td>
</tr>
<tr>
<td>MIB for IPv6: Textual Conventions and General Group</td>
<td>2465</td>
<td>P</td>
</tr>
<tr>
<td>Definitions of MO for Multicast over UNI 3.0/3.1 based ATM Networks</td>
<td>2417</td>
<td>P</td>
</tr>
<tr>
<td>Integrated Services - Guaranteed Service Ext.</td>
<td>2214</td>
<td>P</td>
</tr>
<tr>
<td>Integrated Services</td>
<td>2213</td>
<td>P</td>
</tr>
<tr>
<td>RSVP</td>
<td>2206</td>
<td>P</td>
</tr>
<tr>
<td>IP Forwarding Table</td>
<td>2096</td>
<td>P</td>
</tr>
<tr>
<td>Title</td>
<td>RFC</td>
<td>STATUS</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>IP MIB</td>
<td>2011</td>
<td>P</td>
</tr>
<tr>
<td>IP Mobility Support</td>
<td>2006</td>
<td>P</td>
</tr>
<tr>
<td>OSPF Version 2</td>
<td>1850</td>
<td>D</td>
</tr>
<tr>
<td>RIP Version 2 MIB Extension</td>
<td>1724</td>
<td>D</td>
</tr>
<tr>
<td>BGP Version 4</td>
<td>1657</td>
<td>D</td>
</tr>
<tr>
<td>Identification MIB</td>
<td>1414</td>
<td>P</td>
</tr>
<tr>
<td>BGP Version 3</td>
<td>1269</td>
<td>P</td>
</tr>
<tr>
<td>MIB-II</td>
<td>1213</td>
<td>S</td>
</tr>
</tbody>
</table>
## TRANSPORT LAYER MIBs

<table>
<thead>
<tr>
<th>Title</th>
<th>RFC</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-Time Transport Protocol</td>
<td>2959</td>
<td>P</td>
</tr>
<tr>
<td>IP Version 6 MIB for the User Datagram Protocol</td>
<td>2454</td>
<td>P</td>
</tr>
<tr>
<td>IP Version 6 MIB for the Transmission Control Protocol</td>
<td>2452</td>
<td>P</td>
</tr>
<tr>
<td>User Datagram Protocol (UDP)</td>
<td>2013</td>
<td>P</td>
</tr>
<tr>
<td>Transmission Control Protocol (TCP)</td>
<td>2012</td>
<td>P</td>
</tr>
</tbody>
</table>
# APPLICATION LAYER MIBs

<table>
<thead>
<tr>
<th>Title</th>
<th>RFC</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIB for the PINT Services Architecture</td>
<td>3055</td>
<td>P</td>
</tr>
<tr>
<td>Mail Monitoring MIB</td>
<td>2789</td>
<td>P</td>
</tr>
<tr>
<td>Network Services Monitoring</td>
<td>2788</td>
<td>P</td>
</tr>
<tr>
<td>RADIUS Accounting Server MIB</td>
<td>2621</td>
<td>I</td>
</tr>
<tr>
<td>RADIUS Accounting Client MIB</td>
<td>2620</td>
<td>I</td>
</tr>
<tr>
<td>RADIUS Authentication Server MIB</td>
<td>2619</td>
<td>P</td>
</tr>
<tr>
<td>RADIUS Authentication Client MIB</td>
<td>2618</td>
<td>P</td>
</tr>
<tr>
<td>Directory Server Monitoring MIB</td>
<td>2605</td>
<td>P</td>
</tr>
<tr>
<td>Definitions of Managed Objects for WWW Services</td>
<td>2594</td>
<td>P</td>
</tr>
<tr>
<td>Application Management MIB</td>
<td>2564</td>
<td>P</td>
</tr>
<tr>
<td>Definitions of System-Level Managed Objects for Applications</td>
<td>2287</td>
<td>P</td>
</tr>
<tr>
<td>SNMPv2 MIB</td>
<td>1907</td>
<td>P</td>
</tr>
<tr>
<td>RDBMS MIB</td>
<td>1697</td>
<td>P</td>
</tr>
<tr>
<td>DNS Resolver MIB Extensions</td>
<td>1612</td>
<td>P</td>
</tr>
<tr>
<td>DNS Server MIB Extensions</td>
<td>1611</td>
<td>P</td>
</tr>
</tbody>
</table>
## Remote Monitoring and Measurement

<table>
<thead>
<tr>
<th>Title</th>
<th>RFC</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote Network Monitoring (RMON) MIB</td>
<td>2819</td>
<td>S</td>
</tr>
<tr>
<td>Traffic Flow Measurement: Meter MIB</td>
<td>2720</td>
<td>P</td>
</tr>
<tr>
<td>RMON MIB Extensions for Switched Networks Version 1.0</td>
<td>2613</td>
<td>P</td>
</tr>
<tr>
<td>RMON Version 2</td>
<td>2021</td>
<td>P</td>
</tr>
<tr>
<td>Token Ring extensions to RMON</td>
<td>1513</td>
<td>P</td>
</tr>
</tbody>
</table>
# DISTRIBUTED MANAGEMENT

<table>
<thead>
<tr>
<th>Title</th>
<th>RFC</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notification Log MIB</td>
<td>3014</td>
<td>P</td>
</tr>
<tr>
<td>Expression MIB</td>
<td>2982</td>
<td>P</td>
</tr>
<tr>
<td>Event MIB</td>
<td>2981</td>
<td>P</td>
</tr>
<tr>
<td>Remote Ping, Traceroute, and Lookup Operations</td>
<td>2925</td>
<td>P</td>
</tr>
<tr>
<td>Delegation of Management Scripts</td>
<td>2592</td>
<td>P</td>
</tr>
<tr>
<td>Scheduling Management Operations</td>
<td>2591</td>
<td>P</td>
</tr>
</tbody>
</table>
# VENDOR SPECIFIC MIBs

<table>
<thead>
<tr>
<th>Title</th>
<th>RFC</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPN/HPR in IP Networks</td>
<td>2584</td>
<td>P</td>
</tr>
<tr>
<td>TN3270E Response Time Collection</td>
<td>2562</td>
<td>P</td>
</tr>
<tr>
<td>TN3270E</td>
<td>2561</td>
<td>P</td>
</tr>
<tr>
<td>Extended Border Node</td>
<td>2457</td>
<td>P</td>
</tr>
<tr>
<td>APPN TRAPS</td>
<td>2456</td>
<td>P</td>
</tr>
<tr>
<td>APPN</td>
<td>2455</td>
<td>P</td>
</tr>
<tr>
<td>HPN</td>
<td>2238</td>
<td>P</td>
</tr>
<tr>
<td>DLUR</td>
<td>2232</td>
<td>P</td>
</tr>
<tr>
<td>APPC</td>
<td>2051</td>
<td>P</td>
</tr>
<tr>
<td>TCP/IPX Connection</td>
<td>1792</td>
<td>E</td>
</tr>
<tr>
<td>SNA Data Link Control (SDLC)</td>
<td>1747</td>
<td>P</td>
</tr>
<tr>
<td>Appletalk</td>
<td>1742</td>
<td>P</td>
</tr>
<tr>
<td>SNA NAUs</td>
<td>1666</td>
<td>P</td>
</tr>
<tr>
<td>DECNET Phase IV</td>
<td>1559</td>
<td>D</td>
</tr>
<tr>
<td>Title</td>
<td>RFC</td>
<td>STATUS</td>
</tr>
<tr>
<td>--------------------------------------------------------------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>Common Open Policy Service (COPS) Protocol Clients</td>
<td>2940</td>
<td>P</td>
</tr>
<tr>
<td>Physical Topology</td>
<td>2922</td>
<td>I</td>
</tr>
<tr>
<td>Service Level Agreements Performance Monitoring</td>
<td>2758</td>
<td>E</td>
</tr>
<tr>
<td>Definitions of Managed Objects for Extensible SNMP Agents</td>
<td>2742</td>
<td>P</td>
</tr>
<tr>
<td>Collection and Storage of Accounting Information for CO Networks</td>
<td>2513</td>
<td>P</td>
</tr>
<tr>
<td>Accounting Information for ATM Networks</td>
<td>2512</td>
<td>P</td>
</tr>
<tr>
<td>Textual Conventions for MIB Modules Using Performance History Based on 15 Minute Intervals</td>
<td>2493</td>
<td>P</td>
</tr>
<tr>
<td>Techniques for managing asynchronously generated alerts</td>
<td>1224</td>
<td>E</td>
</tr>
</tbody>
</table>
NAMING OF MIBs

root

ccitt (0)
iso (1)
joint-iso-ccitt (2)

std (0)
reg-auth (1)
mb (2)
org (3)

internet (1)
dod (6)

private (4)
security (5)

snmpV2 (6)
mib-2 (1)

system (1)
interfaces (2)
transmission (10)
snmp (11)

ospf (14)
bgp (15)

... ethernet (6)
token ring (9)
fdi (15)
adsl (94)
...
MIB-II

DEFINES THE VARIABLES TO MANAGE THE TCP/IP PROTOCOL STACK

170 VARIABLES

RFC 1213
SMIv1

ENHANCEMENT OF MIB-I

RFC 1156
DESIGN CRITERIA

• ESSENTIAL FOR FAULT OR CONFIGURATION MANAGEMENT

  • ONLY WEAK CONTROL OBJECTS

  • SMALL NUMBER OF OBJECTS

    • AVOID REDUNDANCY

    • EVIDENCE OF UTILITY

  • DO NOT DISTURB NORMAL OPERATION

  • NO IMPLEMENTATION SPECIFIC ISSUES
MIB-II GROUPS IN A PROTOCOL STACK

SYSTEM

TCP

UDP

IP

ICMP

EGP

AT

INTERFACES

TRANSMISSION

SNMP
NEW VERSIONS

SYSTEM GROUP ⇔ SNMPv2 MIB (RFC 1907)

INTERFACES (IF) GROUP ⇔ IF-MIB (RFC 2863)

ADDRESS TRANSLATION (AT) GROUP ⇔ DEPRECATED

IP & ICMP GROUPS ⇔ IP-MIB (RFC 2011)

TCP GROUP ⇔ TCP-MIB (RFC 2012)

UDP GROUP ⇔ UDP-MIB (RFC 2013)

EGP GROUP ⇔ OUTDATED (BGP)

TRANSMISSION GROUP ⇔ IS PLACEHOLDER

SNMP GROUP ⇔ SNMPv2 MIB (RFC 1907)
SNMPv2 MIB

RFC 1907
PROPOSED STANDARD

• SYSTEM GROUP

• SNMP GROUP

• SNMP MIBObjects GROUP
  snmpTrap
  snmpTraps
  snmpSet (snmpSetSerialNo)
sysServices

physical layer (e.g. repeaters)
data-link layer (e.g. bridges)
internet layer (e.g. IP routers)
end-to-end (e.g. IP Hosts)
application (e.g. nfs-servers)
sysDescr:  "Cisco Gateway"
sysObjectID:  1.3.6.1.4.1.9.1.1
sysUpTime:  37153422  (4 days, 7 h, 12 min, 14.22 s)
sysContact:  "helpdesk@cs.utwente.nl"
sysName:  "utic01.cs.utwente.nl"
sysLocation:  "near logica meeting room"
sysServices:  6  (*bridge and router functions*)
# sysORTable - EXAMPLE

<table>
<thead>
<tr>
<th>sysORIndex</th>
<th>sysORID</th>
<th>sysORDescr</th>
<th>sysORUpTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IF-MIB!ifMIB</td>
<td>The MIB module to describe generic objects for network interface sub-layers</td>
<td>82</td>
</tr>
<tr>
<td>2</td>
<td>SNMPv2-MIB!snmpMIB</td>
<td>The MIB module for SNMPv2 entities</td>
<td>82</td>
</tr>
<tr>
<td>3</td>
<td>TCP-MIB!tcpMIB</td>
<td>The MIB module for managing TCP implementations</td>
<td>82</td>
</tr>
<tr>
<td>4</td>
<td>IP-MIB!ip</td>
<td>The MIB module for managing IP and ICMP implementations</td>
<td>85</td>
</tr>
<tr>
<td>5</td>
<td>UDP-MIB!udpMIB</td>
<td>The MIB module for managing UDP implementations</td>
<td>85</td>
</tr>
</tbody>
</table>
RFC 2863
DRAFT STANDARD

REPLACES IF GROUP OF MIB-II
- RFC 1213
- RFC1229 (EXTENSIONS TO THE IF GROUP)

DEFINES THE FOLLOWING MAIN TABLES:
- ifStackTable
  - ifTable
  - ifXTable
<table>
<thead>
<tr>
<th>ifTable</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>ifIndex</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ifDescr</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ifType</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ifMtu</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ifSpeed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ifPhysAddress</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>ifAdminStatus</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ifOperstatus</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ifLastChange</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ifInOctets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ifInUcastPkts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ifInNUcastPkts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ifInDiscards</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ifInErrors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ifInUnknownProtos</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ifOutOctets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ifOutUcastPkts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ifOutNUcastPkts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ifOutDiscards</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ifOutErrors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ifOutQLen</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ifSpecific</td>
</tr>
</tbody>
</table>
### ifType and ifStatus

#### ifType

**EXAMPLES:**

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Undefined</td>
<td>16</td>
<td>LAPB</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Ethernet</td>
<td>20</td>
<td>ISDN Basic</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>IEEE 802.3</td>
<td>21</td>
<td>ISDN Primary</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>IEEE 802.4</td>
<td>23</td>
<td>PPP</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>IEEE 802.5</td>
<td>24</td>
<td>Loopback</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>IEEE 802.6</td>
<td>28</td>
<td>SLIP</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>FDDI</td>
<td>32</td>
<td>Frame Relay</td>
<td></td>
</tr>
</tbody>
</table>

#### ifAdminStatus / ifOperStatus

- **ifAdminStatus / ifOperStatus**
  - 1 = up
  - 2 = down
  - 3 = testing
IF PACKET COUNT

\[
\begin{align*}
\text{ifInUcastPkts} + \\
\text{ifInNUcastPkts}
\end{align*}
\]

\[
\begin{align*}
\text{ifInDiscards}
\end{align*}
\]

\[
\begin{align*}
\text{ifInUnknownProtos}
\end{align*}
\]

\[
\begin{align*}
\text{ifInErrors}
\end{align*}
\]

\[
\begin{align*}
\text{ifOutUcastPkts} + \\
\text{ifOutNUcastPkts}
\end{align*}
\]

\[
\begin{align*}
\text{ifOutErrors}
\end{align*}
\]

\[
\begin{align*}
\text{ifOutDiscards}
\end{align*}
\]
IP MIB

RFC 2011

- IP GROUP
- ICMP GROUP
- IP MIB Conformance
ipForwarding (1)
ipDefaultTTL (2)
ipInReceives (3)
ipInHdrErrors (4)
ipInAddrErrors (5)
ipInForwDatagrams (7)
ipInUnknownProtos (7)
ipInDiscards (8)
ipInDelivers (9)
ipOutRequest (10)
ipOutDiscards (11)
ipOutNoRoutes (12)
ipReasmTimeout (13)
ipReasmReqds (14)
ipReasmOKs (15)
ipReasmFails (16)
ipFragOKs (17)
ipFragFails (18)
ipFragCreates (19)
ipAddrTable (20)
ipRouteTable (21)
ipNetToMediaTable (22)
ipRoutingDiscards (23)
IP PACKET COUNT

- ipInDelivers
- ipInUnknownProtos
- ipInDiscards
- ipReasmOKs
- ipReasmFails
- ipReasmReqds
- ipInAddrErrors
- ipInHdrErrors
- ipInReceives
- ipForwDatagrams
- ipOutRequests
- ipOutNoRoutes
- ipOutDiscards
- ipFragOKs
- ipFragFails
- ipFragCreates
SNMP PROTOCOL
OVERVIEW OF PDUs

Diagram showing interactions between a manager and an agent involving get, getNext, set, response, and trap operations in the context of MIB management.
MESSAGE & PDU STRUCTURE

**variable bindings:**

<table>
<thead>
<tr>
<th>NAME 1</th>
<th>VALUE 1</th>
<th>NAME 2</th>
<th>VALUE 2</th>
<th>...</th>
<th>...</th>
<th>NAME n</th>
<th>VALUE n</th>
</tr>
</thead>
</table>

**SNMP PDU:**

<table>
<thead>
<tr>
<th>PDU TYPE</th>
<th>REQUEST ID</th>
<th>ERROR STATUS</th>
<th>ERROR INDEX</th>
<th>VARIABLE BINDINGS</th>
</tr>
</thead>
</table>

**SNMP message:**

<table>
<thead>
<tr>
<th>VERSION</th>
<th>COMMUNITY</th>
<th>SNMP PDU</th>
</tr>
</thead>
</table>

TO REQUEST THE VALUE OF 1 OR MORE VARIABLES

POSSIBLE ERRORS:

- **noSuchName** ➞ Object does not exist / Object is not a leaf
  - **tooBig** ➞ Result does not fit in response PDU
  - **genErr** ➞ All other causes
EXAMPLE MIB

1

address (1)
130.89.16.2

info (2)
name (1) uptime (2)
printer-1 123456

route-table (3)
route-entry (1)
dest(1) policy(2) next(3)

<table>
<thead>
<tr>
<th></th>
<th>dest</th>
<th>policy</th>
<th>next</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>
GET EXAMPLES

\[
\text{get}(1.1.0) \\
\text{response}(1.1.0 \Rightarrow 130.89.16.2)
\]

\[
\text{get}(1.2.0) \\
\text{response}(\text{error-status} = \text{noSuchName})
\]

\[
\text{get}(1.1) \\
\text{response}(\text{error-status} = \text{noSuchName})
\]

\[
\text{get}(1.1.0; 1.2.2.0) \\
\text{response}(1.1.0 \Rightarrow 130.89.16.2; 1.2.2.0 \Rightarrow 123456)
\]

\[
\text{get}(1.3.1.3.5.1) \\
\text{response}(1.3.1.3.5.1 \Rightarrow 2)
\]

\[
\text{get}(1.3.1.1.5.1) \\
\text{response}(1.3.1.1.5.1 \Rightarrow 5)
\]

\[
\text{get}(1.3.1.1.5.1, 1.3.1.2.5.1, 1.3.1.3.5.1) \\
\text{response}(1.3.1.1.5.1 \Rightarrow 5, 1.3.1.2.5.1 \Rightarrow 1, 1.3.1.3.5.1 \Rightarrow 2)
\]
TO ASSIGN A VALUE TO AN EXISTING OBJECT INSTANCE

TO CREATE NEW INSTANCES
  • TABLE ROWS

THE SET REQUEST IS ATOMIC

POSSIBLE ERRORS:
  • noSuchName
  • badValue
  • tooBig
  • genErr
SET EXAMPLES

set(1.2.1.0 => my-printer)
response(noError; 1.2.1.0 => my-printer)

set(1.2.1.0 => my-printer, 1.2.2.0 => 0)
response(error-status = noSuchName; error-index = 2)
GET-NEXT

RETRIEVES THE INSTANCE NAME AND VALUE OF THE NEXT MIB ELEMENT TO DISCOVER MIB STRUCTURES TO RETRIEVE TABLE ROWS

POSSIBLE ERRORS:
- noSuchName (= END OF MIB)
- tooBig
- genErr
GET-NEXT EXAMPLES

getNext (1.1.0)
response (1.2.1.0 => printer-1)

getNext (1.2.1.0)
response (1.2.2.0 => 123456)

getNext (1)
response (1.1.0 => 130.89.16.2)

getNext (1.3.1.3.5.1)
response (1.3.1.3.5.2 => 3)

getNext (1.3.1.1; 1.3.1.2; 1.3.1.3)
response (1.3.1.1.2.1 => 2; 1.3.1.2.2.1 => 1; 1.3.1.3.2.1 => 2)

getNext (1.3.1.1.2.1; 1.3.1.2.2.1; 1.3.1.3.2.1)
response (1.3.1.1.3.1 => 3; 1.3.1.2.3.1 => 1; 1.3.1.3.3.1 => 3)
LEXICOGRAPHICAL ORDERING

THE MIB CAN BE CONSIDERED AS AN ORDERED LIST

<table>
<thead>
<tr>
<th>INSTANCE ID</th>
<th>INSTANCE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.0</td>
<td>130.89.16.2</td>
</tr>
<tr>
<td>1.2.1.0</td>
<td>printer-1</td>
</tr>
<tr>
<td>1.2.2.0</td>
<td>123456</td>
</tr>
<tr>
<td>1.3.1.1.2.1</td>
<td>2</td>
</tr>
<tr>
<td>1.3.1.1.3.1</td>
<td>3</td>
</tr>
<tr>
<td>1.3.1.1.5.1</td>
<td>5</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1.3.1.1.9.1</td>
<td>9</td>
</tr>
<tr>
<td>1.3.1.2.2.1</td>
<td>1</td>
</tr>
<tr>
<td>1.3.1.2.3.1</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1.3.1.2.9.1</td>
<td>1</td>
</tr>
<tr>
<td>1.3.1.3.2.1</td>
<td>2</td>
</tr>
<tr>
<td>1.3.1.3.3.1</td>
<td>3</td>
</tr>
<tr>
<td>1.3.1.3.5.1</td>
<td>2</td>
</tr>
<tr>
<td>1.3.1.3.5.2</td>
<td>3</td>
</tr>
<tr>
<td>1.3.1.3.7.1</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
TRAP

TO SIGNAL AN EVENT

TRAP RECEPTION IS NOT CONFIRMED
(THUS UNRELIABLE)

POLLING REMAINS NECESSARY

AGENTS MAY BE CONFIGURED SUCH THAT:
• NO TRAPS WILL BE TRANSMITTED
• TRAPS WILL BE TRANSMITTED TO CERTAIN MANAGERS
SNMPv2

OVERVIEW:

RFCs

LIMITATIONS OF SNMPv1

HISTORY OF SNMPv2
  • HIERARCHIES
  • SECURITY

SNMPv2 PROTOCOL OPERATIONS
SNMPv2 RFCs

COMMUNICATION MODEL
• DRAFT STANDARD
• RFC 1905, RFC1906

SECURITY MODEL - SNMPv2C:
• COMMUNITY BASED SNMP
• SAME ‘SECURITY MECHANISMS’ AS SNMPv1
  • EXPERIMENTAL STATUS
    • RFC 1901

SECURITY MODEL - SNMPv2U:
• USER BASED SECURITY (AUTHENTICATION / ENCRYPTION / ACCESS CONTROL)
  • EXPERIMENTAL STATUS
    • RFC 1909, RFC1910

INFORMATION MODEL:
• STANDARD
• RFC2578, RFC2579, RFC2580
LIMITATIONS OF SNMPv1

• UNDOCUMENTED RULES
• LIMITED ERROR CODES
• LIMITED DATA TYPES
• LIMITED NOTIFICATIONS
• LIMITED PERFORMANCE
• TRANSPORT DEPENDENCE
• LACK OF HIERARCHIES
• LACK OF SECURITY
HISTORY OF SNMPv2

SNMPv2

SNMPv3

DISMAN

SNMP/SMI v1

SMIPv2

SNMP security

SMP


full standard

proposed standard
draft standard
full standard

parties

community

V2Use
V2*
HIERARCHIES: ORIGINAL IDEA

MANAGER TO MANAGER (M2M) MIB

- STANDARD MIB APPROACH
- LIMITED FUNCTIONALITY
- RUN-TIME BEHAVIOUR MUST BE DEFINED AT IMPLEMENTATION TIME
HIERARCHIES: STATUS

WORK HAS MOVED TO A SEPARATE DISTRIBUTED MANAGEMENT GROUP (DISMAN)

THREE APPROACHES ARE STANDARDIZED:

• MIB BASED (EXPRESSION, EVENT AND NOTIFICATION LOG MIB)

  • SCRIPT BASED (SCRIPT AND SCHEDULE MIB)

  • REMOTE OPERATIONS BASED (REMOPS MIB)
SNMPv2 SECURITY: WHAT HAPPENED?

APRIL 1993:
PROPOSED STANDARD
FOUR EDITORS
SECURITY BASED ON PARTIES
FIRST PROTOTYPES APPEARED SOON

JUNE 1995:
PROPOSED STANDARD REJECTED BY TWO OF THE ORIGINAL EDITORS!

AUGUST 1995:
GENERAL AGREEMENT THAT PARTY BASED MODEL WAS TOO COMPLEX!
MANY NEW PROPOSALS APPEARED:
  • SNMPv2C: COMMUNITY BASED
  • SNMPv2U: USER BASED
  • ...

1997:
NEW SNMPv3 WORKING GROUP WAS FORMED
WITH NEW EDITORS
SNMPv2 PROTOCOL OPERATIONS

- **get**
  - manager
  - get
  - response
  - agent

- **getNext**
  - manager
  - getNext
  - response
  - agent

- **set**
  - manager
  - set
  - response
  - agent

- **getBulk**
  - manager
  - getBulk
  - response
  - agent

- **trap**
  - manager
  - trap
  - agent

- **inform**
  - manager
  - inform
  - "agent"
GET-BULK

NEW IN SNMPv2

TO RETRIEVE A LARGE NUMBER OF VARBINDS

IMPROVES PERFORMANCE!
**GET-BULK**

The `getBulk` request has two additional parameters:

- **non-repeators**
- **max-repetitions**

- The first N elements (non-repeators) of the varbind list are treated as if the operation was a normal `getnext` operation.

- The next elements of the varbind list are treated as if the operation consisted of a number (max-repetitions) of repeated `getnext` operations.
GET-BULK

REQUEST\(\text{non-repeaters} = N; \text{max-repetitions} = M;\)
\(\text{VariableBinding-1; ... ; VariableBinding-N; VariableBinding-(N+1); ... ; VariableBinding-(N+R)}\)

RESPONSE\(\)
\(\text{VariableBinding-1; ... ; VariableBinding-N; VariableBinding-(N+1); ... ; VariableBinding-(N+R)}\)
\(\text{1st LEXICOGRAPHICAL SUCCESSOR}\)
\(\text{VariableBinding-(N+1); ... ; VariableBinding-(N+R)}\)
\(\text{2nd LEXICOGRAPHICAL SUCCESSOR}\)
\(\text{VariableBinding-(N+1); ... ; VariableBinding-(N+R)}\)
\(\text{3rd LEXICOGRAPHICAL SUCCESSOR}\)
\(\text{VariableBinding-(N+1); ... ; VariableBinding-(N+R)}\)
\(\text{N-TIMES}\)
\(\text{M-TIMES}\)
\(\text{Mth LEXICOGRAPHICAL SUCCESSOR}\)
GET-BULK EXAMPLE

getBulk(max-repetitions = 4; 1.1)

response(
1.1.0 => 130.89.16.2
1.2.1.0 => printer-1
1.2.2.0 => 123456
1.3.1.1.2.1 => 2)

getBulk(max-repetitions = 3; 1.3.1.1; 1.3.1.2; 1.3.1.3)

response(
1.3.1.1.2.1 => 2; 1.3.1.2.2.1 => 1; 1.3.1.3.2.1 => 2
1.3.1.1.3.1 => 3; 1.3.1.2.3.1 => 1; 1.3.1.3.3.1 => 3
1.3.1.1.5.1 => 5; 1.3.1.2.5.1 => 1; 1.3.1.3.5.1 => 2
)
# SET: NEW ERROR CODES

<table>
<thead>
<tr>
<th>SNMPv1</th>
<th>SNMPv2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PHASE 1:</strong></td>
<td></td>
</tr>
<tr>
<td>badValue</td>
<td>wrongValue</td>
</tr>
<tr>
<td>badValue</td>
<td>wrongEncoding</td>
</tr>
<tr>
<td>badValue</td>
<td>wrongType</td>
</tr>
<tr>
<td>badValue</td>
<td>wrongLength</td>
</tr>
<tr>
<td>badValue</td>
<td>inconsistentValue</td>
</tr>
<tr>
<td>noSuchName</td>
<td>noAccess</td>
</tr>
<tr>
<td>noSuchName</td>
<td>notWritable</td>
</tr>
<tr>
<td>noSuchName</td>
<td>noCreation</td>
</tr>
<tr>
<td>noSuchName</td>
<td>inconsistentName</td>
</tr>
<tr>
<td>noSuchName</td>
<td>resourceUnavailable</td>
</tr>
<tr>
<td>genErr</td>
<td>genErr</td>
</tr>
<tr>
<td>genErr</td>
<td></td>
</tr>
<tr>
<td><strong>PHASE 2:</strong></td>
<td></td>
</tr>
<tr>
<td>genErr</td>
<td>CommitFailed</td>
</tr>
<tr>
<td>genErr</td>
<td>undoFailed</td>
</tr>
</tbody>
</table>
**SNMPv1:**
- COLD START
- WARM START
- LINK DOWN
- LINK UP
- AUTHENTICATION FAILURE
- EGP NEIGHBOR LOSS

**SNMPv2:**
- MIBs MAY NOW INCLUDE NOTIFICATION TYPE MACROS
- FIRST TWO VARBINDS: sysUptime AND snmpTrapOID
- USES SAME FORMAT AS OTHER PDUs
CONFIRMED TRAP

ORIGINALLY TO INFORM A HIGHER LEVEL MANAGER

SAME FORMAT AS TRAP PDU

POSSIBLE ERROR: tooBig
NEW PDU TO SIGNAL PROTOCOL EXCEPTIONS / ERRORS

NO SEMANTICS DEFINED IN SNMPv2
SNMPv3

OVERVIEW:

DESIGN DECISIONS

ARCHITECTURE

SNMP MESSAGE STRUCTURE

SECURE COMMUNICATION
  • USER SECURITY MODEL (USM)

ACCESS CONTROL
  • VIEW BASED ACCESS CONTROL MODEL (VACM)

RFCs
DESIGN DECISIONS

ADDRESS THE NEED FOR SECURITY SET SUPPORT

DEFINE AN ARCHITECTURE THAT ALLOWS FOR LONGEVITY OF SNMP

ALLOW THAT DIFFERENT PORTIONS OF THE ARCHITECTURE MOVES AT DIFFERENT SPEEDS TOWARDS STANDARD STATUS

ALLOW FOR FUTURE EXTENSIONS

KEEP SNMP AS SIMPLE AS POSSIBLE

ALLOW FOR MINIMAL IMPLEMENTATIONS

SUPPORT ALSO THE MORE COMPLEX FEATURES, WHICH ARE REQUIRED IN LARGE NETWORKS

RE-USE EXISTING SPECIFICATIONS, WHENEVER POSSIBLE
SNMPv3 ARCHITECTURE

SNMP ENTITY

SNMP APPLICATIONS

- COMMAND GENERATOR
- COMMAND RESPONDER
- NOTIFICATION ORIGINATOR
- NOTIFICATION RECEIVER
- PROXY FORWARDER
- OTHER

SNMP ENGINE

- DISPATCHER
- MESSAGE PROCESSING SUBSYSTEM
- SECURITY SUBSYSTEM
- ACCESS CONTROL SUBSYSTEM
SNMPv3 ARCHITECTURE: MANAGER

- COMMAND GENERATOR
- NOTIFICATION RECEIVER

- PDU DISPATCHER
  - MESSAGE DISPATCHER
  - TRANSPORT MAPPINGS

- MESSAGE PROCESSING SUBSYSTEM
  - SNMPv1
  - SNMPv2C
  - SNMPv3
  - OTHER

- SECURITY SUBSYSTEM
  - COMMUNITY BASED SECURITY MODEL
  - USER BASED SECURITY MODEL
  - OTHER SECURITY MODEL
SNMPv3 ARCHITECTURE: AGENT

MANAGEMENT INFORMATION BASE

MESSAGE PROCESSING SUBSYSTEM
- SNMPv1
- SNMPv2C
- SNMPv3
- OTHER

ACCESS CONTROL SUBSYSTEM
- VIEW BASED ACCESS CONTROL

SECURITY SUBSYSTEM
- COMMUNITY BASED SECURITY MODEL
- USER BASED SECURITY MODEL
- OTHER SECURITY MODEL

TRANSPORT MAPPINGS

MESSAGE DISPATCHER

PDU DISPATCHER

COMMAND RESPONDER

NOTIFICATION ORIGINATOR
CONCEPTS: snmpEngineID

SNMP ENGINE
snmpEngineID=1

SNMP ENTITY

SNMP ENGINE
snmpEngineID=2

SNMP ENTITY

SNMP ENGINE
snmpEngineID=3

SNMP ENTITY

SNMP ENGINE
snmpEngineID=4

SNMP ENTITY
CONCEPTS: Context

The context can be reached from this engine, thus:
contextEngineID=1
MODULES OF THE SNMPv3 ARCHITECTURE

DISPATCHER AND MESSAGE PROCESSING MODULE
• SNMPv3 MESSAGE STRUCTURE
  • snmpMPDMIB
    • RFC 2572

APPLICATIONS
• snmpTargetMIB
• snmpNotificationMIB
• snmpProxyMIB
  • RFC 2573

SECURITY SUBSYSTEM
• USER BASED SECURITY MODEL
  • snmpUsmMIB
    • RFC 2574

ACCESS CONTROL SUBSYSTEM
• VIEW BASED ACCESS CONTROL MODEL
  • snmpVacmMIB
    • RFC 2574
SNMPv3 MESSAGE STRUCTURE

msgVersion
msgID
msgMaxSize
msgFlags
msgSecurityModel
msgSecurityParameters
customEngineID
customName

PDU

USED BY MESSAGE PROCESSING SUBSYSTEM
USED BY SNMPv3 PROCESSING MODULE
USED BY SECURITY SUBSYSTEM
USED BY ACCESS CONTROL SUBSYSTEM AND APPLICATIONS
SNMPv3 PROCESSING MODULE PARAMETERS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>msgVersion</td>
<td>0..2147483647</td>
</tr>
<tr>
<td>msgID</td>
<td></td>
</tr>
<tr>
<td>msgMaxSize</td>
<td>484..2147483647</td>
</tr>
<tr>
<td>msgFlags</td>
<td></td>
</tr>
<tr>
<td>msgSecurityModel</td>
<td>authFlag, privFlag, reportableFlag</td>
</tr>
<tr>
<td>msgSecurityParameters</td>
<td>SNMPv1, SNMPv2c, USM</td>
</tr>
<tr>
<td>contextEngineID</td>
<td></td>
</tr>
<tr>
<td>contextName</td>
<td></td>
</tr>
</tbody>
</table>

PDU
SECURE COMMUNICATION VERSUS ACCESS CONTROL

MANAGER

APPLICATION PROCESSES

AGENT

MIB

ACCESS CONTROL

SECURE COMMUNICATION

GET / GET-NEXT / GETBULK
SET / TRAP / INFORM

TRANSPORT SERVICE
## USM: SECURITY THREATS

<table>
<thead>
<tr>
<th>THREAT</th>
<th>ADDRESSED?</th>
<th>MECHANISM</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPLAY</td>
<td>YES</td>
<td>TIME STAMP</td>
</tr>
<tr>
<td>MASQUERADE</td>
<td>YES</td>
<td>MD5 / SHA-1</td>
</tr>
<tr>
<td>INTEGRITY</td>
<td>YES</td>
<td>(MD5 / SHA-1)</td>
</tr>
<tr>
<td>DISCLOSURE</td>
<td>YES</td>
<td>DES</td>
</tr>
<tr>
<td>DENIAL OF SERVICE</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>TRAFFIC ANALYSIS</td>
<td>NO</td>
<td></td>
</tr>
</tbody>
</table>
## USM MESSAGE STRUCTURE

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>msgVersion</td>
<td></td>
</tr>
<tr>
<td>msgID</td>
<td></td>
</tr>
<tr>
<td>msgMaxSize</td>
<td></td>
</tr>
<tr>
<td>msgFlags</td>
<td></td>
</tr>
<tr>
<td>msgSecurityModel</td>
<td></td>
</tr>
<tr>
<td>msgAuthoritativeEngineID</td>
<td></td>
</tr>
<tr>
<td>msgAuthoritativeEngineBoots</td>
<td></td>
</tr>
<tr>
<td>msgAuthoritativeEngineTime</td>
<td></td>
</tr>
<tr>
<td>msgUserName</td>
<td></td>
</tr>
<tr>
<td>msgAuthenticationParameters</td>
<td></td>
</tr>
<tr>
<td>msgPrivacyParameters</td>
<td></td>
</tr>
<tr>
<td>contextEngineID</td>
<td></td>
</tr>
<tr>
<td>contextName</td>
<td></td>
</tr>
</tbody>
</table>

### PDU

- **REPLAY**
- **MASQUERADE/INTEGRITY/DISCLOSURE**
- **MASQUERADE/INTEGRITY**
- **DISCLOSURE**
IDEA BEHIND REPLAY PROTECTION

Nonauthoritative Engine

LOCAL NOTION OF REMOTE CLOCK

ID  BOOTS  TIME  DATA

Authoritative Engine

ALLOWED LIFETIME

LOCAL CLOCK

+ \rightarrow ?

ID  BOOTS  TIME  DATA
IDEA BEHIND DATA INTEGRITY AND AUTHENTICATION

ADD THE MESSAGE AUTHENTICATION CODE (MAC) TO THE DATA AND SEND THE RESULT
IDEA BEHIND AUTHENTICATION
IDEA BEHIND THE DATA CONFIDENTIALITY (DES)

DES-KEY

DATA

DES ALGORITHM

ENCRYPTED DATA
IDEA BEHIND ENCRYPTION

DES-KEY
DATA
DES ALGORITHM
ENCRYPTED DATA
USER
ENCRYPTED DATA

DES-KEY
DATA
DES ALGORITHM
ENCRYPTED DATA
USER
ENCRYPTED DATA
VIEW BASED ACCESS CONTROL MODEL

ACCESS CONTROL TABLE

MIB VIEWS
## ACCESS CONTROL TABLES

<table>
<thead>
<tr>
<th>MIB VIEW</th>
<th>ALLOWED OPERATIONS</th>
<th>ALLOWED MANAGERS</th>
<th>REQUIRED LEVEL OF SECURITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Table</td>
<td>SET</td>
<td>John</td>
<td>Authentication Encryption</td>
</tr>
<tr>
<td>Interface Table</td>
<td>GET / GETNEXT</td>
<td>John, Paul</td>
<td>Authentication</td>
</tr>
<tr>
<td>Systems Group</td>
<td>GET / GETNEXT</td>
<td>George</td>
<td>None</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
# SNMPv3 RFCs

<table>
<thead>
<tr>
<th>SNMP ENTITY</th>
<th>RFC 2571</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNMP APPLICATIONS</td>
<td>RFC 2573</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SNMP ENGINE</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 2572</td>
<td>DISPATCHER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RFC 2572</td>
<td>MESSAGE PROCESSING SUBSYSTEM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USM: RFC 2574</td>
<td>SECURITY SUBSYSTEM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VACM: RFC 2575</td>
<td>ACCESS CONTROL SUBSYSTEM</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
EXTENSIBLE AGENTS

FACILITATE THE EXTENSION OF SNMP AGENTS WITH NEW MIB MODULES

- SEPARATE SNMP PROTOCOL ENGINE FROM MIB INSTRUMENTATION

- ALLOW DYNAMIC ADDITION OF NEW MIB MODULE IMPLEMENTATIONS

- EXTENSIBLE AGENTS SHOULD BE TRANSPARENT
**BASIC STRUCTURE**

- **MASTER AGENT**
  - PROTOCOL OPERATIONS
  - ENCODING

- **TRANSPORT**
  - **SUB AGENT**
    - MIB
    - API
  - **SUB AGENT**
    - MIB
    - API
  - **SUB AGENT**
    - MIB
    - API

- **PROTOCOL / IPC**
SPLITTING OF VARBIND LIST

AGENT SYSTEM

MASTER AGENT

A B C

A B

C

SUB AGENT MIB

SUB AGENT MIB

SUB AGENT MIB

MANAGER SYSTEM
CHARACTERISTICS

REQUIRES OID REGISTRATION:

- TOP REGISTRATION
  EXAMPLE: REGISTER(mib-2)

- RANGE REGISTRATION
  EXAMPLE REGISTER(interfaces -> tcp)
POTENTIAL PROBLEMS

• TABLE ENTRIES MAY BE CREATED AND DELETED AT RUN-TIME

• ENTRIES OF A SINGLE TABLE MAY BE LOCATED IN DIFFERENT SUBAGENTS

• DUPLICATED OIDs

• GAPS

• SETS

• sysUpTime
EXAMPLE: GAPS

SUB-AGENT 1

1.1.1.1 1.1.1.2 1.2.1.1 1.2.1.2

SUB-AGENT 2

1.1.2.1 1.1.2.2

GET-NEXT ...
SETS AND ATOMICITY

TRANSACTION-LIKE APPROACH
- TEST
- COMMIT
- UNDO / CLEAN
HISTORY

SMUX (1991: RFC 1227)
SNMP MULTIPLEXING PROTOCOL

DISTRIBUTED PROTOCOL INTERFACE

RESEARCH PROTOTYPES
FOR EXAMPLE: UNIVERSITY OF TWENTE - UT-SNMPv2

COMMERCIAL PRODUCTS
FOR EXAMPLE: SNMP RESEARCH - EMANATE
(ENHANCED MANAGEMENT AGENT THROUGH EXTENSIONS)

AGENTX

PROPOSED IETF STANDARD
  • RFC 2741 & RFC 2742
  • http://www.scguild.com/agentx/

HAS EFFICIENT MESSAGE FORMAT AND CODING

SUPPORTS
  • SUBAGENTS IMPLEMENTING SEPARATE MIB MODULES
  • SUBAGENTS IMPLEMENTING ROWS IN "SIMPLE TABLES"
  • SUBAGENTS SHARING TABLES ALONG NON-ROW BORDERS

NON-GOALS
  • SUBAGENTS SHARING "COMPLEX TABLES"
  • SUBAGENT TO SUBAGENT COMMUNICATION
AGENTX - NORMAL PDUS

- Get
  - master
- GetNext
  - master
- GetBulk
  - master
- TestSet
  - master
- CommitSet
  - master
- UndoSet
  - master
- CleanupSet
  - master
- Notify
  - master
**EXAMPLE: PDU FORMAT OF GetNext**

<table>
<thead>
<tr>
<th>VERSION</th>
<th>TYPE</th>
<th>FLAGS</th>
<th>RESERVED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SESSION ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRANSACTION ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PACKET ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PAYLOAD LENGTH</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CONTEXT (OPTIONAL)**

<table>
<thead>
<tr>
<th>LENGTH</th>
<th>PREFIX</th>
<th>INCLUDE</th>
<th>RESERVED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FIRST SUB IDENTIFIER</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LAST SUB IDENTIFIER</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OBJECT 1</th>
<th>START OF RANGE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJECT 1</td>
<td>END OF RANGE</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OBJECT N</th>
<th>START OF RANGE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJECT 1</td>
<td>END OF RANGE</td>
<td></td>
</tr>
</tbody>
</table>
AGENTX - ADMINISTRATIVE PDUS

Open
Close

AddAgentCaps
RemoveAgentCaps

Register
Unregister

IndexAllocate
IndexDeallocate

Ping

Response
TO ESTABLISH A SESSION

A UNIQUE sessionID IS ASSIGNED

SUBAGENT SPECIFIES DEFAULT TIME-OUT

RESPONSES FROM MASTER ALWAYS INCLUDE sysUpTime

SESSION CAN BE CLOSED BY MASTER OR SUBAGENT
AGENT CAPABILITIES

TO INFORM THE MASTER OF THE AGENT’S CAPABILITIES

CAPABILITIES ARE DEFINED AS:

- AN OBJECT ID
- A HUMAN READABLE STRING

THE CAPABILITIES ARE STORED IN THE sysORTable
REGISTRATION

CHOICE BETWEEN:
• TOP REGISTRATION
• RANGE REGISTRATION

PRIORITY CAN BE SPECIFIED
• TO DETERMINE THE AUTHORITATIVE SUBAGENT

TIME-OUT CAN BE SPECIFIED
INDEXALLOCATION

TO ALLOCATE ONE OR MORE TABLE ROWS

SUBAGENT REQUESTS ALLOCATION OF:
  • A SPECIFIC INDEX VALUE
  • AN INDEX VALUE THAT IS NOT CURRENTLY ALLOCATED
  • AN INDEX VALUE THAT HAS NEVER BEEN ALLOCATED

MASTER AGENT MAINTAINS DATABASE

AFTER ALLOCATION REGISTRATION IS STILL NEEDED
TO MONITOR IF THE MASTER AGENT IS STILL ABLE TO RECEIVE AND SEND AGENTX PDUs
DISTRIBUTED MANAGEMENT

THREE APPROACHES ARE BEING DEFINED

MIB BASED
  • EXPRESSION MIB
  • EVENT MIB
  • NOTIFICATION LOG MIB

SCRIPT BASED
  • SCRIPT MIB
  • SCHEDULE MIB

REMOTE OPERATIONS BASED
  • REMOTE OPERATIONS MIB
EXPRESSION AND EVENT MIB

TOP LEVEL MANAGER

INTERMEDIATE LEVEL MANAGER

AGENT
EXPRESSION AND EVENT MIB: CHARACTERISTICS

• STANDARD MIB APPROACH

• RESEMBLES THE OLD SNMPv2 M2M MIB

EXPRESSION MIB:
• INPUT ARE (WILDCARDED) VARIABLES OF A (LOCAL) MIB
• OPERATES ON ABSOLUTE AS WELL AS DELTA VALUES
  • RICH SET OF EXPRESSIONS
  • THE OUTPUT IS STORED IN THE VALUE TABLE
• THIS TABLE MAY SERVE AS INPUT FOR OTHER EXPRESSIONS

EVENT MIB:
• INPUT ARE VARIABLES OF A (REMOTE) MIB
• TRIGGERS ON CHANGES, OR TRESHOLD CROSSING
• GENERATES A NOTIFICATION OR SET OPERATION
SCRIPT MIB

SCRIPT MIB

HTTP
SNMP
ICMP

M
SCRIPT REPOSITORY

TOP LEVEL MANAGER

INTERMEDIATE LEVEL MANAGER

SCRIPT MIB
RUNTIME ENVIRONMENT
SNMP
HTTP
ICMP
SNMP

AGENTS
A A A A A A
SCRIPT MIB: CHARACTERISTICS

• FUNCTIONALITY CAN BE DEFINED AT RUN-TIME

• POWERFUL AUTONOMOUS ACTIONS

• MAY BE EASIER TO OPERATE FOR THE TOP-LEVEL MANAGER

• PROTECTION MECHANISMS NECESSARY

• DIFFERENT SCRIPT LANGUAGES
SCRIPT MIB: STRUCTURE

CONSISTS OF 6 TABLES
SCRIPT MIB: LANGUAGE TABLE

Defines the languages this system supports

- An OID to indicate the language
- The version
- An OID to indicate the vendor
- The revision
- A description

Table is read only
SCRIPT MIB: EXTENSIONS TABLE

DEFINES THE EXTENSIONS FOR EACH LANGUAGE

- AN OID TO INDICATE THE EXTENSION
  - THE VERSION
- AN OID TO INDICATE THE VENDOR
  - THE REVISION
  - A DESCRIPTION

TABLE IS READ ONLY
SCRIPT MIB: SCRIPT TABLE

LISTS ALL SCRIPTS KNOWN TO THE SYSTEM

ALLOWS TO:

- DOWNLOAD SCRIPTS FROM A URL (PULL MODEL)
- READ SCRIPTS FROM LOCAL NON-VOLATILE STORAGE
- STORE SCRIPTS IN LOCAL NON-VOLATILE STORAGE
- DELETE SCRIPTS FROM LOCAL NON-VOLATILE STORAGE
- LIST PERMANENT SCRIPTS (THAT CAN NOT BE CHANGED OR REMOVED)
- READ AND MODIFY THE SCRIPT STATUS (ENABLED, DISABLED, EDITING)
SCRIPT MIB: CODE TABLE

LISTS THE CODE OF A SCRIPT

ALLOWS TO:

- DOWNLOAD SCRIPTS VIA SNMP (PUSH MODEL)
- MODIFY SCRIPTS VIA SNMP (EDITING)

IMPLEMENTATION IS OPTIONAL
SCRIPT MIB: LAUNCH TABLE

ALLOWS TO:

• ASSOCIATE A SCRIPT WITH A ‘PERSON’ WHO INVOKES EXECUTION
• PROVIDE ARGUMENTS AND PARAMETERS FOR SCRIPT INVOCATION
  • INVOKE SCRIPTS WITH A SINGLE SET OPERATION
• CONTROL THE NUMBER OF ACTIVE INVOCATIONS
• CONTROL THE TOTAL NUMBER OF INVOCATIONS
ALLOWS TO:

- RETRIEVE STATUS INFORMATION FROM RUNNING SCRIPTS
- CONTROL RUNNING SCRIPTS (SUSPEND, RESUME, ABORT)
- RETRIEVE RESULTS FROM RECENTLY TERMINATED SCRIPTS
- CONTROL THE REMAINING MAXIMUM LIFETIME OF A RUNNING SCRIPT
- CONTROL HOW LONG SCRIPT RESULTS ARE ACCESSIBLE
SCHEDULE MIB

PERFORMS SET OPERATIONS

FOR EXAMPLE ON THE SCRIPT MIB
• TARGET MUST BE Integer32

ON A PERIODIC OR CALENDER DRIVEN BASE
REMOTE OPERATIONS MIB

PING MIB
• TO PERFORM PING FROM A REMOTE HOST

TRACEROUTE MIB
• TO PERFORM TRACEROUTE FROM A REMOTE HOST

NAME LOOKUP MIB
• TO PERFORM NAME LOOKUP FROM A REMOTE HOST
NEW DEVELOPMENTS

POLICY BASED MANAGEMENT

BANDWIDTH BROKER

LDAP?

POLICY REPOSITORY

COPS? / SNMP?

PEP

ROUTER

PEP

ROUTER

PEP

ROUTER

PDP

POLICY DECISION POINT
COPS VERSUS SNMP

COPS:
• SPECIAL CASE OF CONFIGURATION MANAGEMENT
• HIGHER LEVEL OBJECTS THAN USUAL WITH SNMP
  • POLICY INFORMATION BASE (PIB)
• SINGLE OPERATION TO ADD OR DELETE TABLE ROWS
• RELIABLE COMMUNICATION BETWEEN PDP AND PEP (BECAUSE OF TCP)
  • EACH PEP IS CONNECTED TO SINGLE PDP

SNMP:
• INTEGRATED APPROACH TO MANAGEMENT
• POLICIES CAN BE DEFINED WITHIN MIBs
• EACH PEP MAY BE CONNECTED TO MULTIPLE PDPs
WWW SERVERS

- IETF
  http://www.ietf.org/

- The SimpleWeb
  http://www.simpleweb.org/

- The Simple Times
  http://www.simple-times.org/

- The Smurfland NM Web Server
  http://netman.cit.buffalo.edu/
BOOKS

• W. Stallings
SNMP, SNMPv2, SNMPv3 and RMON1 and 2
Third edition, Addison-Wesley, 1999
ISBN: 0-201-48534-6

• D. Zeltserman
A Practical Guide to SNMPv3 and Network Management
Prentice Hall, 1999
ISBN: 0-13-021453-1

• D. Perkins, E. McGinnis
Understanding SNMP MIBs
Prentice Hall, 1996
ISBN: 0-13-437708-7
ARTICLES

The Simple Times: Special issue on Agent Extensibility
Issue 4-2, April 1996

The Simple Times: Special issue on SNMPv3
Issue 5-1, December 1997

The Simple Times: An overview of the AgentX Protocol
Issue 6-1, March 1998

The Simple Times: Special issue on SNMPv3
Issue 7-2, November 1999

William Stallings,
Security Comes to SNMP: The New SNMPv3 Proposed Internet Standards

William Stallings,
SNMPv3: A Security Enhancement for SNMP,
IEEE Communications Survey, Q4, 1998