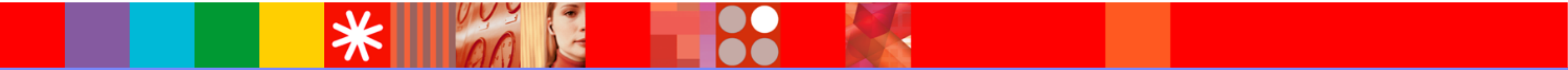
IBM Software Group

SNMP 101

Michael Shannon

Tivoli software



ON DEMAND BUSINESS

How We Got Here

- Developed in 1988
- Reaction to CMIP, which was designed for telecom devices
  - Emphasized ‘simple’
  - Security deliberately not included
- Interim protocol to allow growth of internet

SNMP Versions

- V1 - 1988
  - Basic structure is still in use today
- V2 - 1993
  - Introduced new security model
  - Not widely used
- V2c - 1996
  - Continued V1’s security model
    - V2 security model was seen as too complex and confusing
  - Introduced getBulk
  - PDU formats slightly different (esp. Traps)
  - Introduced Informs (alternative to Traps)
- V3 - 2002
  - New security model

Terms

- Agent
  - The software entity that responds to SNMP requests; the managed device or software
- ASN.1
  - Abstract syntax notation 1 - language used to define SNMP objects
- BER
  - Basic Encoding Rules – method of serializing data based on type, length and value
- Manager
  - The software entity that issues SNMP requests; the managing device or software
- MIB
  - Management Information Base – a set of information maintained by an SNMP agent
- PDU
  - Protocol Data Unit – an SNMP message
- SMI
  - Structure of Management Information – the public MIB that defines the overall structure, common data types, and information that should be common to all agents

SNMP PDUs – V1

- GetRequest
  - Used to retrieve a single value
- GetNextRequest
  - Used to retrieve the next available value
  - Generally used to ‘walk’ columns in tables
- GetResponse
  - PDU used for a response to a GetRequest, GetNextRequest or SetRequest
- SetRequest
  - Used to change configuration of the agent
- Trap
  - Asynchronous notification from the agent
  - Uses a different format

SNMP PDUs – V2c and V3

- GetRequest
  - Used to retrieve a single value
- GetNextRequest
  - Used to retrieve the next available value
  - Generally used to walk columns in tables
- GetResponse
  - PDU used for a response to a GetRequest, GetNextRequest or SetRequest
- SetRequest
  - Used to change configuration of the agent
- SNMPV2Trap
  - Same format as other PDUs
- InformRequest
  - Similar to SNMPV2Trap
- Report
  - Not currently defined

SNMP PDUs – V2c and V3 (continued)

- GetBulkRequest
  - Retrieves larger volumes of data
  - Can retrieve values for multiple rows of a table in one request
  - Non-Repeaters field
    - How many scalar values are being requested
  - Max-Repetitions field
    - How many rows (maximum) should be returned for the remaining variables
  - Better way to ‘walk’ a table

MIBs

- Tree-structured database
- Provided in MIB files
- Written in ASN.1
- Specification allows both public, standard MIBs and private, enterprise MIBs
  - Public MIBs defined in RFCs
    - MIB-II defines information that should be common to all agents
  - Private MIBs under the enterprise portion of the tree
    - Companies may choose whether to publish their MIBs
- Structure of Management Information contains basic definitions
  - Data types (e.g. Integer)/Textual Conventions
  - Public portion of tree structure
- IANA (Internet Assigned Numbers Authority) is responsible for assigning enterprise numbers

MIBs

- Object Identifier
  - Identifies branches in the tree
  - Does not represent a value
- Object Type
  - Table or Scalar
  - Represents a value (or values)
  - May or may not be able to retrieve

MIB Objects – Types

- Simple Types
  - INTEGER, OCTET STRING, OBJECT IDENTIFIER, NULL
  - 32 and 64 bit versions introduced with V2
- Structured Types
  - SEQUENCE OF
    - Defines tables
  - SEQUENCE
    - Defines the row of a table
- Defined Types
  - IpAddress, Display String, Counter, Gauge, TimeTicks, Opaque
  - Textual Conventions (introduced with V2)

MIB – File Header

```
SNMPv2-MIB DEFINITIONS ::= BEGIN

   IMPORTS
       MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
       TimeTicks, Counter32, snmpModules, mib-2
           FROM SNMPv2-SMI
       DisplayString, TestAndIncr, TimeStamp
           FROM SNMPv2-TC
       MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
           FROM SNMPv2-CONF;

   snmpMIB MODULE-IDENTITY
       LAST-UPDATED "200210160000Z"
       ORGANIZATION "IETF SNMPv3 Working Group"
       CONTACT-INFO
               "WG-EMail:   snmpv3@lists.tislabs.com
                Subscribe:  snmpv3-request@lists.tislabs.com

                Co-Chair:   Russ Mundy
                            Network Associates Laboratories
                postal:     15204 Omega Drive, Suite 300
                            Rockville, MD 20850-4601
                            USA
                EMail:      mundy@tislabs.com
                phone:      +1 301 947-7107
(…)
```

MIB – File Header (continued)

```
(…)
DESCRIPTION
               "The MIB module for SNMP entities.

                Copyright (C) The Internet Society (2002). This
                version of this MIB module is part of RFC 3418;
                see the RFC itself for full legal notices.
               "
       REVISION      "200210160000Z"
       DESCRIPTION
               "This revision of this MIB module was published as
                RFC 3418."
       REVISION      "199511090000Z"
       DESCRIPTION
               "This revision of this MIB module was published as
                RFC 1907."
       REVISION      "199304010000Z"
       DESCRIPTION
               "The initial revision of this MIB module was published
               as RFC 1450."
       ::= { snmpModules 1 }
```

MIB – Object Identifiers

- Identifies a branch of the tree
- No value associated
  - Can not be queried
  - Can not be further defined

```
system   OBJECT IDENTIFIER ::= { mib-2 1 }
```

MIB Objects – Scalars

- Single value variables
- May use any of the Simple or Defined types
- When queried, they are identified by an instance of ‘0’
  - Example – sysUpTime.0 = 14571

```
sysDescr OBJECT-TYPE
      SYNTAX      DisplayString (SIZE (0..255))
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
              "A textual description of the entity.  This value
  should
              include the full name and version identification of
              the system's hardware type, software operating-system,
              and networking software."
      ::= { system 1 }
```

MIB Objects – Scalars (continued)

```
sysObjectID OBJECT-TYPE
      SYNTAX      OBJECT IDENTIFIER
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
              "The vendor's authoritative identification of the
              network management subsystem contained in the entity.
              This value is allocated within the SMI enterprises
              subtree (1.3.6.1.4.1) and provides an easy and
              unambiguous means for determining `what kind of box' is
              being managed.  For example, if vendor `Flintstones,
              Inc.' was assigned the subtree 1.3.6.1.4.1.424242,
              it could assign the identifier 1.3.6.1.4.1.424242.1.1
              to its `Fred Router'."
      ::= { system 2 }
```

MIB Objects – Scalars (continued)

```
sysContact OBJECT-TYPE
      SYNTAX      DisplayString (SIZE (0..255))
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
              "The textual identification of the contact person for
              this managed node, together with information on how
              to contact this person.  If no contact information is
              known, the value is the zero-length string."
      ::= { system 4 }
```

MIB Objects – Scalars (continued)

```
sysServices OBJECT-TYPE
      SYNTAX      INTEGER (0..127)
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
              "A value which indicates the set of services that this
              entity may potentially offer.  The value is a sum.
              This sum initially takes the value zero. Then, for
              each layer, L, in the range 1 through 7, that this node
              performs transactions for, 2 raised to (L - 1) is added
              to the sum.  For example, a node which performs only
              routing functions would have a value of 4 (2^(3-1)).
              In contrast, a node which is a host offering application
              services would have a value of 72 (2^(4-1) + 2^(7-1)).
              Note that in the context of the Internet suite of
              protocols, values should be calculated accordingly:

                   layer      functionality
                     1        physical (e.g., repeaters)
                     2        datalink/subnetwork (e.g., bridges)
                     3        internet (e.g., supports the IP)
                     4        end-to-end  (e.g., supports the TCP)
                     7        applications (e.g., supports the SMTP)

              For systems including OSI protocols, layers 5 and 6
              may also be counted."
      ::= { system 7 }
```

MIB Objects – Scalars (continued)

```
snmpEnableAuthenTraps OBJECT-TYPE
      SYNTAX      INTEGER { enabled(1), disabled(2) }
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
              "Indicates whether the SNMP entity is permitted to
              generate authenticationFailure traps.  The value of this
              object overrides any configuration information; as such,
              it provides a means whereby all authenticationFailure
              traps may be disabled.

              Note that it is strongly recommended that this object
              be stored in non-volatile memory so that it remains
              constant across re-initializations of the network
              management system."
      ::= { snmp 30 }
```

MIB Objects – Textual Conventions

- Define new types
- Often used to limit size
- Also can introduce enumerations
- Allows reuse of definitions
  - TCs can be imported from other MIB files

```
MacAddress ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "1x:"
    STATUS       current
    DESCRIPTION
            "Represents an 802 MAC address represented in the
            `canonical' order defined by IEEE 802.1a, i.e., as if it
            were transmitted least significant bit first, even though
            802.5 (in contrast to other 802.x protocols) requires MAC
            addresses to be transmitted most significant bit first."
    SYNTAX       OCTET STRING (SIZE (6))
```

MIB Objects – Textual Conventions (continued)

```
TruthValue ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
            "Represents a boolean value."
    SYNTAX       INTEGER { true(1), false(2) }

VariablePointer ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
            "A pointer to a specific object instance.  For example,
            sysContact.0 or ifInOctets.3."
    SYNTAX       OBJECT IDENTIFIER
```

MIB Objects – Textual Conventions (continued)

```
RowStatus ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
            "The RowStatus textual convention is used to manage the
            creation and deletion of conceptual rows, and is used as the
            value of the SYNTAX clause for the status column of a
            conceptual row (as described in Section 7.7.1 of [2].)
(…)
   SYNTAX       INTEGER {
                     -- the following two values are states:
                     -- these values may be read or written
                     active(1),
                     notInService(2),

                     -- the following value is a state:
                     -- this value may be read, but not written
                     notReady(3),

                     -- the following three values are
                     -- actions: these values may be written,
                     --   but are never read
                     createAndGo(4),
                     createAndWait(5),
                     destroy(6)
                 }
```

MIB Objects – Textual Conventions (continued)

```
DateAndTime ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "2d-1d-1d,1d:1d:1d.1d,1a1d:1d"
    STATUS       current
    DESCRIPTION
            "A date-time specification.

            field  octets  contents                  range
            -----  ------  --------                  -----
              1      1-2   year*                     0..65536
              2       3    month                     1..12
              3       4    day                       1..31
              4       5    hour                      0..23
              5       6    minutes                   0..59
              6       7    seconds                   0..60
                           (use 60 for leap-second)
              7       8    deci-seconds              0..9
              8       9    direction from UTC        '+' / '-'
              9      10    hours from UTC*           0..13
             10      11    minutes from UTC          0..59

            * Notes:
            - the value of year is in network-byte order
            - daylight saving time in New Zealand is +13

            For example, Tuesday May 26, 1992 at 1:30:15 PM EDT would be
            displayed as:

                             1992-5-26,13:30:15.0,-4:0

            Note that if only local time is known, then timezone
            information (fields 8-10) is not present."
    SYNTAX       OCTET STRING (SIZE (8 | 11))
```

MIB Objects – Sequences and Tables

- Three parts to define a table
  - Table Object Identifier
    - Defined as a SEQUENCE of a defined type
  - Entry Object Identifier
    - Begins with lower case
    - Defined as a defined type
    - Identifies the index variable(s)
  - Entry Sequence
    - Defined as a Sequence
- Then individual fields are defined

MIB Objects – Sequences and Tables (Table Object Identifier)

```
ifTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF IfEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "A list of interface entries.  The number of entries is
            given by the value of ifNumber."
    ::= { interfaces 2 }
```

MIB Objects – Sequences and Tables (Entry Object Identifier)

```
ifEntry OBJECT-TYPE
    SYNTAX      IfEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "An entry containing management information applicable to a
            particular interface."
    INDEX   { ifIndex }
    ::= { ifTable 1 }
```

MIB Objects – Sequences and Tables (Entry Sequence)

```
IfEntry ::=
    SEQUENCE {
        ifIndex                 InterfaceIndex,
        ifDescr                 DisplayString,
        ifType                  IANAifType,
        ifMtu                   Integer32,
        ifSpeed                 Gauge32,
        ifPhysAddress           PhysAddress,
        ifAdminStatus           INTEGER,
        ifOperStatus            INTEGER,
        ifLastChange            TimeTicks,
        ifInOctets              Counter32,
        ifInUcastPkts           Counter32,
        ifInNUcastPkts          Counter32,  -- deprecated
        ifInDiscards            Counter32,
        ifInErrors              Counter32,
        ifInUnknownProtos       Counter32,
        ifOutOctets             Counter32,
        ifOutUcastPkts          Counter32,
        ifOutNUcastPkts         Counter32,  -- deprecated
        ifOutDiscards           Counter32,
        ifOutErrors             Counter32,
        ifOutQLen               Gauge32,    -- deprecated
        ifSpecific              OBJECT IDENTIFIER -- deprecated
    }
```

MIB Objects – Sequences and Tables (Individual Fields)

```
ifIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "A unique value, greater than zero, for each interface.  It
            is recommended that values are assigned contiguously
            starting from 1.  The value for each interface sub-layer
            must remain constant at least from one re-initialization of
            the entity's network management system to the next re-
            initialization."
    ::= { ifEntry 1 }
```

MIB Objects – Sequences and Tables (Individual Fields - continued)

```
ifDescr OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..255))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "A textual string containing information about the
            interface.  This string should include the name of the
            manufacturer, the product name and the version of the
            interface hardware/software."
    ::= { ifEntry 2 }

ifType OBJECT-TYPE
    SYNTAX      IANAifType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The type of interface.  Additional values for ifType are
            assigned by the Internet Assigned Numbers Authority (IANA),
            through updating the syntax of the IANAifType textual
            convention."
    ::= { ifEntry 3 }
```

MIB Objects – Sequences and Tables (Individual Fields - continued)

```
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 }
```

MIB Objects – Sequences and Tables (Individual Fields - continued)

```
ifOperStatus OBJECT-TYPE
    SYNTAX  INTEGER {
                up(1),          -- ready to pass packets
                down(2),
                testing(3),     -- in some test mode
                unknown(4),     -- status can not be determined
                                -- for some reason.
                dormant(5),
                notPresent(6),      -- some component is missing
                lowerLayerDown(7) -- down due to state of
                                  -- lower-layer interface(s)
            }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The current operational state of the interface.  The
            testing(3) state indicates that no operational packets can
            be passed.  If ifAdminStatus is down(2) then ifOperStatus
            should be down(2).  If ifAdminStatus is changed to up(1)
            then ifOperStatus should change to up(1) if the interface is
            ready to transmit and receive network traffic; it should
            change to dormant(5) if the interface is waiting for
            external actions (such as a serial line waiting for an
            incoming connection); it should remain in the down(2) state
            if and only if there is a fault that prevents it from going
            to the up(1) state; it should remain in the notPresent(6)
            state if the interface has missing (typically, hardware)
            components."
    ::= { ifEntry 8 }
```

MIB Objects – Notifications

- V1 – Traps
  - No acknowledgement
  - PDU format different from other operations
    - Agent address in Trap PDU
    - Enterprise (OID)
    - Generic Trap type
      - 6, enterpriseSpecific, used for most traps
    - Specific Trap type
    - Variable Bindings for additional information
- V2 – Traps
  - No acknowledgement
  - PDU format the same as other operations
    - No agent address field in PDU
    - Single field contains Enterprise, Generic Trap type and Specific Trap type
    - Variable Bindings for additional information

MIB Objects – Notifications (continued)

- V2/V3 – InformRequest
  - Requires acknowledgement
  - PDU format the same as other operations
    - No agent address field in PDU
    - Single field contains Enterprise, Generic Trap type and Specific Trap type
    - Variable Bindings for additional information

MIB Objects – Notifications (continued)

- Six Generic Trap Types
  - coldStart
    - Agent has reinitialized – configuration may have changed
    - Usually power cycle
  - warmStart
    - Agent has reinitialized – configuration has not changed
  - linkDown
    - Communication link has failed
  - linkUp
    - Communication link has recovered
  - authenticationFailure
    - Agent has received an SNMP PDU with an incorrect community string
  - egpNeighborLoss
    - Not generally used; specific to an obsolete routing protocol
  - enterpriseSpecific
    - Trap is identified by enterprise and specific trap type

MIB Objects – Notifications (continued)

- Generic Trap Type 0 (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
```

MIB Objects – Notifications (continued)

- Same trap redefined in V2

```
coldStart NOTIFICATION-TYPE
       STATUS  current
       DESCRIPTION
               "A coldStart trap signifies that the SNMP entity,
               supporting a notification originator application, is
               reinitializing itself and that its configuration may
               have been altered."
       ::= { snmpTraps 1 }
```

MIB Objects – Notifications (continued)

- Enterprise specific trap

```
sipCommonMIBNotifications OBJECT IDENTIFIER ::= { sipCommonMIB 0 }
(…)
sipCommonStatusCodeNotif NOTIFICATION-TYPE
    OBJECTS {
       sipCommonNotifSequenceNumber,
       sipCommonNotifApplIndex,
       sipCommonStatusCodeNotifTo,
       sipCommonStatusCodeNotifFrom,
       sipCommonStatusCodeNotifCallId,
       sipCommonStatusCodeNotifCSeq,
       sipCommonStatusCodeIns,
       sipCommonStatusCodeOuts
    }
    STATUS      current
    DESCRIPTION
       "Signifies that a specific status code has been sent or received
        by the system."
    ::= { sipCommonMIBNotifications 1 }
```

MIB Objects – Conformance

- Marks which parts of the MIB must be supported, and which are optional
- Generally not useful for SNMP consumers
  - Companies may change SNMP support
  - May or may not change conformance clauses to match