Network Working Group
Request for Comments: 1565
Category: Standards Track

S. Kille, WG Chair
ISODE Consortium
N. Freed, Editor
Innosoft

January 1994

Network Services Monitoring MIB

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Table of Contents

1. Introduction

There are a wide range of networked applications for which it is appropriate to provide SNMP Monitoring. This includes both TCP/IP and OSI applications. This document defines a MIB which contains the elements common to the monitoring of any network service application. This information includes a table of all monitorable network service applications, a count of the associations (connections) to each application, and basic information about the parameters and status of each application-related association.

This MIB may be used on its own for any application, and for most simple applications this will suffice. This MIB is also designed to serve as a building block which can be used in conjunction with application-specific monitoring and management. Two examples of this are MIBs defining additional variables for monitoring a Message Transfer Agent (MTA) service or a Directory Service Agent (DSA) service. It is expected that further MIBs of this nature will be specified.

This MIB does not attempt to provide facilities for management of the host or hosts the network service application runs on, nor does it provide facilities for monitoring applications that provide something other than a network service. Host resource and general application monitoring is handled by the Host Resources MIB.

2. The SNMPv2 Network Management Framework

The SNMPv2 Network Management Framework consists of four major components. They are:

- RFC 1442 [1] which defines the SMI, the mechanisms used for describing and naming objects for the purpose of management.
- STD 17, RFC 1213 [2] defines MIB-II, the core set of managed objects for the Internet suite of protocols.
- RFC 1445 [3] which defines the administrative and other architectural aspects of the framework.
- RFC 1448 [4] which defines the protocol used for network access to managed objects.

The Framework permits new objects to be defined for the purpose of experimentation and evaluation.

2.1 Object Definitions

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the subset of Abstract Syntax Notation One (ASN.1) defined in the SMI. In particular, each object type is named by an OBJECT IDENTIFIER, an administratively assigned name. The object type together with an object instance serves to uniquely identify a specific instantiation of the object. For human convenience, we often use a textual string, termed the descriptor, to refer to the object type.

3. Rationale for having a Network Services Monitoring MIB

Much effort has been expended in developing tools to manage lower layer network facilities. However, relatively little work has been done on managing application layer entities. It is neither efficient nor reasonable to manage all aspects of application layer entities using only lower layer information. Moreover, the difficulty of managing application entities in this way increases dramatically as application entities become more complex.

This leads to a substantial need to monitor applications which provide network services, particularly distributed components such as MTAs and DSAs, by monitoring specific aspects of the application itself. Reasons to monitor such components include but are not limited to measuring load, detecting broken connectivity, isolating system failures, and locating congestion.

In order to manage network service applications effectively two requirements must be met:

(1) It must be possible to monitor a large number of components (typical for a large organization).

(2) Application monitoring must be integrated into general network management.

This specification defines simple read-only access; this is sufficient to determine up/down status and provide an indication of a broad class of operational problems.

3.1 General Relationship to Other MIBs

This MIB is intended to only provide facilities common to the monitoring of any network service application. It does not provide all the facilities necessary to monitor any specific application. Each specific type of network service application is expected to have a MIB of its own that makes use of these common facilities.

3.2 Restriction of Scope

The framework provided here is very minimal; there is a lot more that could be done. For example:

(1) General network service application configuration monitoring and control.

(2) Detailed examination and modification of individual entries in service-specific request queues.

(3) Probing to determine the status of a specific request (e.g. the location of a mail message with a specific message-id).

(4) Requesting that certain actions be performed (e.g. forcing an immediate connection and transfer of pending messages to some specific system).

All these capabilities are both impressive and useful. However, these capabilities would require provisions for strict security checking. These capabilities would also mandate a much more complex design, with many characteristics likely to be fairly implementation-specific. As a result such facilities are likely to be both contentious and difficult to implement.

This document religiously keeps things simple and focuses on the basic monitoring aspect of managing applications providing network services. The goal here is to provide a framework which is simple, useful, and widely implementable.

3.3 Relationship to Directory Services

Use of and management of directory services already is tied up with network service application management. There are clearly many things which could be dealt with by directory services and protocols. We take the line here that static configuration information is both provided by and dealt with by directory services and protocols. The emphasis here is on transient application status.

By placing static information in the directory, the richness and linkage of the directory information framework does not need to be repeated in the MIB. Static information is information which has a mean time to change of the order of days or longer.

When information about network service applications is stored in the directory (regardless of whether or not the network service application makes direct use of the directory), it is recommended that a linkage be established, so that:

(1) The managed object contains its own directory name. This allows all directory information to be obtained by reference. This will let a SNMP monitor capable of performing directory queries present this information to the manager in an appropriate format. It is intended that this will be the normal case.

(2) The directory will reference the location of the SNMP agent, so that an SNMP capable directory query agent could probe dynamic characteristics of the object.

(3) This approach could be extended further, so that the SNMP attributes are modelled as directory attributes. This would dramatically simplify the design of directory service agents that use SNMP to obtain the information they need.

4. Application Objects

This MIB defines a set of general purpose attributes which would be appropriate for a range of applications that provide network services. Both OSI and non-OSI services can be accomodated. Additional tables defined in extensions to this MIB provide attributes specific to specific network services.

A table is defined which will have one row for each network service application running on the system. The only static information held on the application is its name. All other static information should be obtained from various directory services. The applDirectoryName is an external key, which allows an SNMP MIB entry to be cleanly related to the X.500 Directory. In SNMP terms, the applications are grouped in a table called applTable, which is indexed by an integer key applIndex.

The type of the application will be determined by one or both of:

(1) Additional MIB variables specific to the applications.

(2) An association to the application of a specific protocol.

5. Definitions

```
   APPLICATION-MIB DEFINITIONS ::= BEGIN

   IMPORTS
       OBJECT-TYPE, Counter32, Gauge32
         FROM SNMPv2-SMI
       mib-2
         FROM RFC1213-MIB
       DisplayString, TimeStamp
         FROM SNMPv2-TC;


   -- Textual conventions

   -- DistinguishedName [5] is used to refer to objects in the
   -- directory.

   DistinguishedName ::= TEXTUAL-CONVENTION
       STATUS current
       DESCRIPTION
           "A Distinguished Name represented in accordance with
            RFC1485."
       SYNTAX DisplayString

   application MODULE-IDENTITY
       LAST-UPDATED "9311280000Z"
       ORGANIZATION "IETF Mail and Directory Management Working Group"
       CONTACT-INFO
         "        Ned Freed

          Postal: Innosoft International, Inc.
                  250 West First Street, Suite 240
                  Claremont, CA  91711
                  US

             Tel: +1 909 624 7907
             Fax: +1 909 621 5319

          E-Mail: ned@innosoft.com"
       DESCRIPTION
         "The MIB module describing network service applications"
       ::= { mib-2 27 }

   -- The basic applTable contains a list of the application
   -- entities.
```

```
   applTable OBJECT-TYPE
       SYNTAX SEQUENCE OF ApplEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
           "The table holding objects which apply to all different
            kinds of applications providing network services."
       ::= {application 1}

   applEntry OBJECT-TYPE
       SYNTAX ApplEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
         "An entry associated with a network service application."
       INDEX {applIndex}
       ::= {applTable 1}

   ApplEntry ::= SEQUENCE {
       applIndex
           INTEGER,
       applName
           DisplayString,
       applDirectoryName
           DistinguishedName,
       applVersion
           DisplayString,
       applUptime
           TimeStamp,
       applOperStatus
           INTEGER,
       applLastChange
           TimeStamp,
       applInboundAssociations
           Gauge32,
       applOutboundAssociations
           Gauge32,
       applAccumulatedInboundAssociations
           Counter32,
       applAccumulatedOutboundAssociations
           Counter32,
       applLastInboundActivity
           TimeStamp,
       applLastOutboundActivity
           TimeStamp,
       applRejectedInboundAssociations
           Counter32,
       applFailedOutboundAssociations
```

```
        Counter32
  }

  applIndex OBJECT-TYPE
      SYNTAX INTEGER (1..2147483647)
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "An index to uniquely identify the network service
         application."
      ::= {applEntry 1}

  applName OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "The name the network service application chooses to be
         known by."
      ::= {applEntry 2}

  applDirectoryName OBJECT-TYPE
      SYNTAX DistinguishedName
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "The Distinguished Name of the directory entry where
         static information about this application is stored.
         An empty string indicates that no information about
         the application is available in the directory."
      ::= {applEntry 3}

  applVersion OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "The version of network service application software."
      ::= {applEntry 4}
```

```
   applUptime OBJECT-TYPE
       SYNTAX TimeStamp
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
         "The value of sysUpTime at the time the network service
          application was last initialized.  If the application was
          last initialized prior to the last initialization of the
          network management subsystem, then this object contains
          a zero value."
       ::= {applEntry 5}

   applOperStatus OBJECT-TYPE
       SYNTAX INTEGER {
         up(1),
         down(2),
         halted(3),
         congested(4),
         restarting(5)
       }
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
         "Indicates the operational status of the network service
          application. 'down' indicates that the network service is
          not available. 'running' indicates that the network service
          is operational and available.  'halted' indicates that the
          service is operational but not available.  'congested'
          indicates that the service is operational but no additional
          inbound associations can be accomodated.  'restarting'
          indicates that the service is currently unavailable but is
          in the process of restarting and will be available soon."
       ::= {applEntry 6}

   applLastChange OBJECT-TYPE
       SYNTAX TimeStamp
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
         "The value of sysUpTime at the time the network service
          application entered its current operational state.  If
          the current state was entered prior to the last
          initialization of the local network management subsystem,
          then this object contains a zero value."
       ::= {applEntry 7}
```

```
   applInboundAssociations OBJECT-TYPE
       SYNTAX Gauge32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
         "The number of current associations to the network service
          application, where it is the responder.  For dynamic single
          threaded processes, this will be the number of application
          instances."
       ::= {applEntry 8}

   applOutboundAssociations OBJECT-TYPE
       SYNTAX Gauge32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
         "The number of current associations to the network service
          application, where it is the initiator.  For dynamic single
          threaded processes, this will be the number of application
          instances."
       ::= {applEntry 9}

   applAccumulatedInboundAssociations OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
         "The total number of associations to the application entity
          since application initialization, where it was the responder.
          For  dynamic single threaded processes, this will be the
          number of application instances."
       ::= {applEntry 10}

   applAccumulatedOutboundAssociations OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
         "The total number of associations to the application entity
          since application initialization, where it was the initiator.
          For dynamic single threaded processes, this will be the
          number of application instances."
       ::= {applEntry 11}
```

```
   applLastInboundActivity OBJECT-TYPE
       SYNTAX TimeStamp
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
         "The value of sysUpTime at the time this application last
          had an inbound association.  If the last association
          occurred prior to the last initialization of the network
          subsystem, then this object contains a zero value."
       ::= {applEntry 12}

   applLastOutboundActivity OBJECT-TYPE
       SYNTAX TimeStamp
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
         "The value of sysUpTime at the time this application last
          had an outbound association.  If the last association
          occurred prior to the last initialization of the network
          subsystem, then this object contains a zero value."
       ::= {applEntry 13}

   applRejectedInboundAssociations OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
         "The total number of inbound associations the application
          entity has rejected, since application initialization."
       ::= {applEntry 14}

   applFailedOutboundAssociations OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
         "The total number associations where the application entity
          is initiator and association establishment has failed,
          since application initialization."
       ::= {applEntry 15}


   -- The assocTable augments the information in the applTable
   -- with information about associations.  Note that two levels
   -- of compliance are specified below, depending on whether
   -- association monitoring is mandated.
```

```
   assocTable OBJECT-TYPE
       SYNTAX SEQUENCE OF AssocEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
           "The table holding a set of all active application
            associations."
       ::= {application 2}

   assocEntry OBJECT-TYPE
       SYNTAX AssocEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
         "An entry associated with an association for a network
          service application."
       INDEX {applIndex, assocIndex}
       ::= {assocTable 1}

   AssocEntry ::= SEQUENCE {
       assocIndex
           INTEGER,
       assocRemoteApplication
           DisplayString,
       assocApplicationProtocol
           OBJECT IDENTIFIER,
       assocApplicationType
           INTEGER,
       assocDuration
           TimeStamp
   }

   assocIndex OBJECT-TYPE
       SYNTAX INTEGER (1..2147483647)
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
         "An index to uniquely identify each association for a network
          service application."
       ::= {assocEntry 1}
```

```
   assocRemoteApplication OBJECT-TYPE
       SYNTAX DisplayString
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
         "The name of the system running remote network service
          application.  For an IP-based application this should be
          either a domain name or IP address.  For an OSI application
          it should be the string encoded distinguished name of the
          managed object.  For X.400(84) MTAs which do not have a
          Distinguished Name, the RFC1327 [6] syntax
          'mta in globalid' should be used."
       ::= {assocEntry 2}

   assocApplicationProtocol OBJECT-TYPE
       SYNTAX OBJECT IDENTIFIER
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
         "An identification of the protocol being used for the
          application.  For an OSI Application, this will be the
          Application Context.  For Internet applications, the IANA
          maintains a registry of the OIDs which correspond to
          well-known applications.  If the application protocol is
          not listed in the registry, an OID value of the form
          {applTCPProtoID port} or {applUDPProtoID port} are used for
          TCP-based and UDP-based protocols, respectively. In either
          case 'port' corresponds to the primary port number being
          used by the protocol."
       ::= {assocEntry 3}

   assocApplicationType OBJECT-TYPE
       SYNTAX INTEGER {
           ua-initiator(1),
           ua-responder(2),
           peer-initiator(3),
           peer-responder(4)}
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
         "This indicates whether the remote application is some type of
          client making use of this network service (e.g. a User Agent)
          or a server acting as a peer. Also indicated is whether the
          remote end initiated an incoming connection to the network
          service or responded to an outgoing connection made by the
          local application."
       ::= {assocEntry 4}
```

```
   assocDuration OBJECT-TYPE
       SYNTAX TimeStamp
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
         "The value of sysUpTime at the time this association was
          started.  If this association started prior to the last
          initialization of the network subsystem, then this
          object contains a zero value."
       ::= {assocEntry 5}


   -- Conformance information

   applConformance OBJECT IDENTIFIER ::= {application 3}

   applGroups      OBJECT IDENTIFIER ::= {applConformance 1}
   applCompliances OBJECT IDENTIFIER ::= {applConformance 2}


   -- Compliance statements

   applCompliance MODULE-COMPLIANCE
       STATUS current
       DESCRIPTION
         "The compliance statement for SNMPv2 entities
          which implement the Network Services Monitoring MIB
          for basic monitoring of network service applications."
       MODULE  -- this module
         MANDATORY-GROUPS {applGroup}
       ::= {applCompliances 1}

   assocCompliance MODULE-COMPLIANCE
       STATUS current
       DESCRIPTION
         "The compliance statement for SNMPv2 entities which
          implement the Network Services Monitoring MIB for basic
          monitoring of network service applications and their
          associations."
       MODULE  -- this module
         MANDATORY-GROUPS {applGroup, assocGroup}
       ::= {applCompliances 2}
```

```
   -- Units of conformance

   applGroup OBJECT-GROUP
       OBJECTS {
         applName, applVersion, applUptime, applOperStatus,
         applLastChange, applInboundAssociations,
         applOutboundAssociations, applAccumulatedInboundAssociations,
         applAccumulatedOutboundAssociations, applLastInboundActivity,
         applLastOutboundActivity, applRejectedInboundAssociations,
         applFailedOutboundAssociations}
       STATUS current
       DESCRIPTION
         "A collection of objects providing basic monitoring of
          network service applications."
       ::= {applGroups 1}

   assocGroup OBJECT-GROUP
       OBJECTS {
         assocRemoteApplication, assocApplicationProtocol,
         assocApplicationType, assocDuration}
       STATUS current
       DESCRIPTION
         "A collection of objects providing basic monitoring of
          network service applications' associations."
       ::= {applGroups 2}


   -- OIDs of the form {applTCPProtoID port} are intended to be used
   -- for TCP-based protocols that don't have OIDs assigned by other
   -- means. {applUDPProtoID port} serves the same purpose for
   -- UDP-based protocols. In either case 'port' corresponds to
   -- the primary port number being used by the protocol. For example,
   -- assuming no other OID is assigned for SMTP, an OID of
   -- {applTCPProtoID 25} could be used, since SMTP is a TCP-based
   -- protocol that uses port 25 as its primary port.

   applTCPProtoID OBJECT IDENTIFIER ::= {application 4}
   applUDPProtoID OBJECT IDENTIFIER ::= {application 5}

   END
```

6. Acknowledgements

This document is a product of the Mail and Directory Management (MADMAN) Working Group. It is based on an earlier MIB designed by S. Kille, T. Lenggenhager, D. Partain, and W. Yeong.

7. References

[1] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1442, SNMP Research, Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.

[2] McCloghrie, K., and M. Rose, Editors, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", STD 17, RFC 1213, Hughes LAN Systems, Performance Systems International, March 1991.

[2] Galvin, J., and K. McCloghrie, "Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1445, Trusted Information Systems, Hughes LAN Systems, April 1993.

[4] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1448, SNMP Research, Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.

[5] Kille, S., "A String Representation of Distinguished Names", RFC 1485, ISODE Consortium, July 1993.

[6] Kille, S., "Mapping between X.400(1988) / ISO 10021 and RFC822", RFC 1327, University College London, May 1992.

8. Security Considerations

Security issues are not discussed in this memo.

Authors' Addresses

Steve Kille, WG Chair
ISODE Consortium
The Dome, The Square
Richmond TW9 1DT
UK

Phone: +44 81 332 9091
EMail: S.Kille@isode.com

Ned Freed, Editor
Innosoft International, Inc.
250 West First Street, Suite 240
Claremont, CA 91711
USA

Phone: +1 909 624 7907
Fax: +1 909 621 5319
EMail: ned@innosoft.com