

Network Working Group
Request for Comments: 4498
Category: Experimental

G. Keeni
Cyber Solutions Inc.
May 2006

The Managed Object Aggregation MIB

Status of This Memo

This memo defines an Experimental Protocol for the Internet community. It does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2006).

IESG Note

The content of this RFC was at one time considered by the IETF, and therefore it may resemble a current IETF work in progress or a published IETF work. This RFC is not a candidate for any level of Internet Standard. The IETF disclaims any knowledge of the fitness of this RFC for any purpose and in particular notes that the decision to publish is not based on IETF review for such things as security, congestion control, or inappropriate interaction with deployed protocols. The RFC Editor has chosen to publish this document at its discretion. Readers of this RFC should exercise caution in evaluating its value for implementation and deployment. See RFC 3932 for more information.

Abstract

This memo defines a portion of the Management Information Base (MIB), the Aggregation MIB modules, for use with network management protocols in the Internet community. In particular, the Aggregation MIB modules will be used to configure a network management agent to aggregate the values of a user-specified set of Managed Object instances and to service queries related to the aggregated Managed Object instances.

Table of Contents

1. The Internet-Standard Management Framework	2
2. Background	2
3. MO Aggregation: The Concept	3
4. The Requirements for Managed Object Aggregation	6
5. MIB Design	6
6. The Aggregation MIB Modules	7
7. Security Considerations	25
8. IANA Considerations	27
9. References	27
9.1. Normative References	27
9.2. Informative References	27
10. Acknowledgements	28

1. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP).

Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIV2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

2. Background

For the purpose of management, it is necessary to access Managed Objects (MOs). The SNMP framework provides a mechanism for naming and describing managed objects. These objects are accessed via a virtual information store termed a Management Information Base (MIB). MIBs have been defined by equipment, protocol, and application developers to provide management access to the managed entities. We will call the MOs defined in these MIBs simple MOs (SMO). Management applications will access one or more instances of these SMOs, one or more times, to monitor the target entity.

There is a cost associated with accessing MOs. The cost is the network bandwidth and the packet header processing overhead at the command generator (manager) and the command responder (agent). This cost constrains the number of MO instances that can be polled and the interval at which polling can be carried out.

The overhead reduction can be carried out by reducing the number of query-response packets. This will reduce the packet processing overhead, and to some extent, the bandwidth.

The payloads in a typical SNMP "get" packet and the corresponding response are as shown in Figure 1. In this example, polling is carried out for 'n' Managed Object instances OID1, OID2, ..., OIDn. It is obvious that a substantial amount of the payload in an SNMP packet consists of the OIDs.

3. MO Aggregation: The Concept

In this document, a mechanism of MO aggregation for payload compression is defined. The idea is simple: we introduce the concept of an Aggregate MO (AgMO). An AgMO is just another MO as far as the SNMP protocol is concerned. No new protocol operations will be required to handle these MOs. As in the case of any other MO, it requires additional instrumentation at the command responder (agent) and at the (command generator) manager. In this mechanism, the user defines an Aggregate MO (AgMO) corresponding to one or more (predefined) MO instances. Semantically, the value of an AgMO instance will be equivalent to the concatenation of the values of the corresponding MO instances. The order of the concatenation will be determined by the order in which the MO instances are specified in the AgMO definition. With the definitions done, the user can, as and when the necessity arises, do an SNMP 'get' on instances of the AgMO to fetch the value of the constituent MO instances. There is substantial savings on bandwidth, as only one instance object identifier is carried in the request and the response. In the normal case, instance object identifiers for each of the constituent MO instances would be carried in the requests and the responses. This is the basic concept of Aggregate Managed Objects. For every AgMO, an ErrorStatus Managed Object is defined. This MO indicates errors, if any, that have been encountered while fetching the values of the constituent MO instances. The error indication is comprised of the index of the MO instance and the corresponding error. If there are no errors, the ErrorStatus Managed Object instance will have a null value. This is the basic concept of Aggregate Managed Objects.

The concepts are explained in Figure 2. An aggregate managed object, AgMOx, has been defined for the MO instances MOI1, ... MOIn. The value of an instance of AgMOx will be a concatenation of the values of MOI1, ... MOIn, in that order.

Polling for MO Instances [MOI1, MOI2, ... MOIn]:

```

Query:  +-----+-----+-----+... -+-----+-----+
        |Get req | MOI1 | NULL  |   | MOIn | NULL  |
        +-----+-----+-----+... -+-----+-----+

Response: +-----+-----+-----+... -+-----+-----+
          |Get resp| MOI1 | Val1  |   | MOIn | Valn  |
          +-----+-----+-----+... -+-----+-----+

```

Figure 1. Polling for MO instances

Polling for an instance (AgMOIx) of an aggregate MO (AgMOx):

AgMOx = aggr{AgMOI1, AgMOI2,AgMOIn}

```

Query:  +-----+-----+-----+
        |Get req | AgMOIx | NULL  |
        +-----+-----+-----+

Response: +-----+-----+-----+-----+
          |Get resp| AgMOIx |   Val1,Val2,...,Valn   |
          +-----+-----+-----+-----+

```

Figure 2. MO aggregation

As a further refinement of the AgMO, we introduce the Time-Based Aggregated Managed Object (TAgMO). The TAgMO is an MO that represents the values of a user-specified MO instance sampled at user-specified intervals for a user-specified number of times. In this case, the user defines a TAgMO by specifying the MO instance that needs to be sampled, the sampling interval, and the desired number of samples that will be included in one TAgMO. The value of a TAgMO instance will include the timestamp (sysUpTime) at which the first sample was taken. The start time is not specified when the TAgMO is defined. Implementations may choose to align the start time with the appropriate time boundaries (e.g., seconds, minutes, hours). With the definitions, the user can do an SNMP "get" on an instance of the TAgMO to fetch the values of the constituent MO instance sampled at the specified intervals. This is the concept of Time-Based aggregation.

Polling for 'n' samples of an MO Instance [MOI] at an interval 'i':

Query =====	Time =====	Response =====
+-----+-----+-----+ Get req MOI NULL	t	
+-----+-----+-----+	:	+-----+-----+-----+
	:	Get resp MOI Val(t)
	:	+-----+-----+-----+
+-----+-----+-----+ Get req MOI NULL	t+i	
+-----+-----+-----+	:	+-----+-----+-----+
	:	Get resp MOI Val(t+i)
	X	+-----+-----+-----+
	X	
	:	
+-----+-----+-----+ Get req MOI NULL	t+(n-1)i	
+-----+-----+-----+	:	+-----+-----+-----+
	:	Get resp MOI Val(t+(n-1)i)
	:	+-----+-----+-----+

Figure 3. Periodic polling for samples of an MO instance

Polling for an instance (TAgMOIx) of a Time-Based aggregate MO (TAgMOx):

TAgMOx = aggr{'n' polled samples of an instance (MOI) of MO
at intervals = 'i' microseconds}

Query:	+-----+-----+-----+ Get req TAgMOIx NULL +-----+-----+-----+
Response:	+-----+-----+-----+ Get resp TAgMOIx t,Val(t),Val(t+i),..,Val(t + (n-1)*i) +-----+-----+-----+

Figure 4. Time-Based aggregation

The TAgMO instance is a "bucket" of data representing the value of the corresponding MO instance sampled at 'i' microsecond intervals, 'n' times (i.e., over a 'n' X 'i' microsecond window). The TAgMO instance value gets updated at 'n' X 'i' microsecond intervals.

4. The Requirements for Managed Object Aggregation

The general requirements of managed object aggregation are as follows:

- o It should lead to fewer packets.
- o It should lead to less bandwidth consumption.
- o It should not lead to loss of information.

In the case of Time-Based aggregation, there will be a delay involved in getting the actual data. The minimum delay in this case will be the duration of the aggregation.

The manager application is expected to configure AgMOs (Aggregate MOs) and TAgMOs (Time-Based Aggregate MOs) with care so that the response size is not too large. In case the resultant response size is larger than the maximum acceptable message size of the originator or larger than the local maximum message size, then the error-status field will be set to "tooBig".

Note that an aggregate MO can be defined only when all the constituent MO instances of interest are known. This scheme cannot be employed if a manager/application does not know the specific MO instances (of interest) that are serviced by the management target. In such cases, the application may "discover" the MO instances of interest by some means, e.g., by "walking" through the MIB tree on the agent. According to the results of the "walk", the application can define an appropriate aggregate MO that will serve the purpose. Considering the cost involved in this exercise, this method is recommended only if the aggregate MO will be used repeatedly, so that the benefits of aggregation outweigh the costs of configuration.

5. MIB Design

The basic principle has been to keep the MIB as simple as possible and at the same time to make it flexible enough that a large number of users and applications can use the MIB to configure aggregate MOs conveniently.

Two separate MIB modules have been defined. The AggrMIB supports the aggregation of independent MO instances, while TAggrMIB supports the aggregation of several samples of the same MO instance. Both of these MIB modules use the textual conventions defined in RMON-MIB [RFC2819] and SNMP-FRAMEWORK-MIB [RFC3411].

The AggrMIB is comprised of three tables, described below.

- The aggrCtlTable controls the aggregation process. Each row in this table defines the attributes of the aggregate object defined in the aggrMOTable.
- The aggrMOTable defines the primary MO-based aggregation, i.e., the MOs that will be aggregated.
- The aggrDataTable contains the details of the aggregated object.

The TAggrMIB is comprised of two tables described below.

- The tAggrCtlTable controls the aggregation process. Each row in this table defines the attributes of the aggregate object defined in the aggrMOTable.
- The tAggrDataTable contains the details of the aggregated object.

6. The Aggregation MIB Modules

```
AGGREGATE-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    MODULE-IDENTITY, experimental, Unsigned32,
```

```
    OBJECT-TYPE, Opaque
```

```
        FROM SNMPv2-SMI
```

```
    OwnerString
```

```
        FROM RMON-MIB
```

```
    RowStatus, StorageType, TEXTUAL-CONVENTION
```

```
        FROM SNMPv2-TC
```

```
    MODULE-COMPLIANCE, OBJECT-GROUP
```

```
        FROM SNMPv2-CONF
```

```
    SnmpAdminString
```

```
        FROM SNMP-FRAMEWORK-MIB;
```

```
aggrMIB MODULE-IDENTITY
```

```
    LAST-UPDATED "200604270000Z"          -- 27th April, 2006
```

```
    ORGANIZATION "Cyber Solutions Inc. NetMan Working Group"
```

```
    CONTACT-INFO
```

```
    "
```

```
        Glenn Mansfield Keeni
```

```
        Postal: Cyber Solutions Inc.
```

```
        6-6-3, Minami Yoshinari
```

```
        Aoba-ku, Sendai, Japan 989-3204.
```

```
        Tel: +81-22-303-4012
```

```
        Fax: +81-22-303-4015
```

```
        E-mail: glenn@cysols.com
```

```
        Support Group E-mail: mibsupport@cysols.com"
```

DESCRIPTION

"The MIB for servicing aggregate objects.

Copyright (C) The Internet Society (2006). This version of this MIB module is part of RFC 4498; see the RFC itself for full legal notices.

"

REVISION "200604270000Z" -- 27th April, 2006

DESCRIPTION "Initial version, published as RFC 4498."
 ::= { experimental 123 }

AggrMOErrorStatus ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"This data type is used to model the error status of the constituent MO instances. The error status for a constituent MO instance is given in terms of two elements:

- o The moIndex, which indicates the position of the MO instance (starting at 1) in the value of the aggregated MO instance.
- o The moError, which indicates the error that was encountered in fetching that MO instance.

The syntax in ASN.1 Notation will be

```
ErrorStatus ::= SEQUENCE {
    moIndex  Integer32,
    moError  SnmpPduErrorStatus
}
```

```
AggrMOErrorStatus ::= SEQUENCE OF {
    ErrorStatus
}
```

Note1: The command responder will supply values for all constituent MO instances, in the same order in which the MO instances are specified for the AgMO. If an error is encountered for an MO instance, then the corresponding value will have an ASN.1 value NULL, and an error will be flagged in the corresponding AggrMOErrorStatus object. Only MOs for which errors have been encountered will have their corresponding moIndex and moError values set.

Note2: The error code for the component MO instances will be in accordance with the SnmpPduErrorStatus TC defined in the DISMAN-SCHEDULE-MIB [RFC3231].

Note3: The command generator will need to know constituent MO instances and their order to correctly interpret AggrMOErrorStatus.

"

SYNTAX Opaque (SIZE (0..1024))


```

AggrMOValue ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "This data type is used to model the aggregate
        MOs. It will have a format dependent on the constituent
        MOs, a sequence of values. The syntax in ASN.1 Notation will
        be
        MOValue ::= SEQUENCE {
            value ObjectSyntax
        }
        where 'value' is the value of a constituent MO instance.
        AggrMOValue ::= SEQUENCE OF {
            MOValue
        }

```

Note: The command generator will need to know the constituent MO instances and their order to correctly interpret AggrMOValue."

```

SYNTAX      Opaque (SIZE (0..1024))

```

```

AggrMOCompressedValue ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "This data type is used to model the compressed
        aggregate MOs."
    SYNTAX      OCTET STRING (SIZE (0..1024))

```

```

--
-- The aggregation control table
-- There will be a row for each aggregate MO
--

```

```

aggrCtlTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF AggrCtlEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that controls the aggregation of the MOs."
    ::= {aggrMIB 1}

```

```

aggrCtlEntry OBJECT-TYPE
    SYNTAX      AggrCtlEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A row of the control table that defines one aggregated
        MO.

```

Entries in this table are required to survive a reboot of the managed entity depending on the value of the corresponding `aggrCtlEntryStorageType` instance.

"

```
INDEX {aggrCtlEntryID }
::= {aggrCtlTable 1 }
```

```
AggrCtlEntry ::= SEQUENCE {
    aggrCtlEntryID
        SnmpAdminString,
    aggrCtlMOIndex
        Unsigned32,
    aggrCtlMODescr
        SnmpAdminString,
    aggrCtlCompressionAlgorithm
        INTEGER,
    aggrCtlEntryOwner
        OwnerString,
    aggrCtlEntryStorageType
        StorageType,
    aggrCtlEntryStatus
        RowStatus
}
```

```
aggrCtlEntryID OBJECT-TYPE
    SYNTAX SnmpAdminString (SIZE(1..32))
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "A locally unique, administratively assigned name
        for this aggregated MO. It is used as an index to
        uniquely identify this row in the table."
    ::= { aggrCtlEntry 1 }
```

```
aggrCtlMOIndex OBJECT-TYPE
    SYNTAX Unsigned32 (1..2147483647)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "A pointer to a group of MOs identified by aggrMOEntryID
        in the aggrMOTable. This is the group of MOs that will
        be aggregated."
    ::= { aggrCtlEntry 2 }
```

```
aggrCtlMODescr OBJECT-TYPE
    SYNTAX SnmpAdminString (SIZE(0..64))
    MAX-ACCESS read-create
    STATUS current
```

```
DESCRIPTION
    "A textual description of the object that is
    being aggregated."
    ::= {aggrCtlEntry 3}

-- only one compression algorithm is defined as of now.
aggrCtlCompressionAlgorithm OBJECT-TYPE
    SYNTAX INTEGER {
        none          (1),
        deflate       (2)
    }
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The compression algorithm that will be used by
        the agent to compress the value of the aggregated
        object.
        The deflate algorithm and corresponding data format
        specification is described in RFC 1951. It is
        compatible with the widely used gzip utility.
        "
    REFERENCE
        "RFC1951 : DEFLATE Compressed Data Format Specification
        version 1.3
        "
    DEFVAL { none }
    ::= {aggrCtlEntry 4}

aggrCtlEntryOwner OBJECT-TYPE
    SYNTAX OwnerString
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The entity that created this entry."
    ::= {aggrCtlEntry 5}

aggrCtlEntryStorageType OBJECT-TYPE
    SYNTAX StorageType
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This object defines whether the parameters defined in
        this row are kept in volatile storage and lost upon
        reboot or backed up by non-volatile (permanent)
        storage.

        Conceptual rows having the value 'permanent' need not
        allow write-access to any columnar objects in the row.
```

```

"
 ::= {aggrCtlEntry 6}

aggrCtlEntryStatus OBJECT-TYPE
    SYNTAX RowStatus
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The row status variable, used according to row
        installation and removal conventions.
        Objects in a row can be modified only when the value of
        this object in the corresponding conceptual row is not
        'active'.
        Thus, to modify one or more of the objects in this
        conceptual row,
            a. change the row status to 'notInService',
            b. change the values of the row, and
            c. change the row status to 'active'.
        The aggrCtlEntryStatus may be changed to 'active' if
        all the MOs in the conceptual row have been assigned
        valid values.
"
 ::= {aggrCtlEntry 7}

--
-- The Table of primary(simple) MOs
--
aggrMOTable OBJECT-TYPE
    SYNTAX SEQUENCE OF AggrMOEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The table of primary(simple) MOs that will be aggregated.
        Each row in this table represents a MO that will be
        aggregated. The aggrMOEntryID index is used to identify
        the group of MOs that will be aggregated. The
        aggrMOIndex instance in the corresponding row of the
        aggrCtlTable will have a value equal to the value of
        aggrMOEntryID. The aggrMOEntryMOID index is used to
        identify an MO in the group.
"
 ::= {aggrMIB 2}

aggrMOEntry OBJECT-TYPE
    SYNTAX AggrMOEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION

```

```
"A row of the table that specifies one MO.
Entries in this table are required to survive a reboot
of the managed entity depending on the value of the
corresponding aggrMOEntryStorageType instance.
"
INDEX { aggrMOEntryID, aggrMOEntryMOID }
::= { aggrMOTable 1 }

AggrMOEntry ::= SEQUENCE {
    aggrMOEntryID
        Unsigned32,
    aggrMOEntryMOID
        Unsigned32,
    aggrMOInstance
        OBJECT IDENTIFIER,
    aggrMODescr
        SnmpAdminString,
    aggrMOEntryStorageType
        StorageType,
    aggrMOEntryStatus
        RowStatus
}

aggrMOEntryID OBJECT-TYPE
    SYNTAX Unsigned32 (1..2147483647)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An index uniquely identifying a group of MOs
        that will be aggregated."
    ::= { aggrMOEntry 1 }

aggrMOEntryMOID OBJECT-TYPE
    SYNTAX Unsigned32 (1..65535)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An index to uniquely identify an MO instance in the
        group of MO instances that will be aggregated."
    ::= { aggrMOEntry 2 }

aggrMOInstance OBJECT-TYPE
    SYNTAX OBJECT IDENTIFIER
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The OID of the MO instance, the value of which will
        be sampled by the agent."
```

```
 ::= { aggrMOEntry 3 }

aggrMODescr OBJECT-TYPE
    SYNTAX SnmpAdminString (SIZE(0..64))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "A textual description of the object that will
        be aggregated."
    ::= { aggrMOEntry 4}

aggrMOEntryStorageType OBJECT-TYPE
    SYNTAX StorageType
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This object defines whether the parameters defined in
        this row are kept in volatile storage and lost upon
        reboot or backed up by non-volatile (permanent)
        storage.
        Conceptual rows having the value 'permanent' need not
        allow write-access to any columnar objects in the row.
        "
    ::= { aggrMOEntry 5}

aggrMOEntryStatus OBJECT-TYPE
    SYNTAX RowStatus
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The row status variable, used according to row
        installation and removal conventions.
        Objects in a row can be modified only when the value of
        this object in the corresponding conceptual row is not
        'active'.
        Thus, to modify one or more of the objects in this
        conceptual row,
            a. change the row status to 'notInService',
            b. change the values of the row, and
            c. change the row status to 'active'.
        The aggrMOEntryStatus may be changed to 'active' iff
        all the MOs in the conceptual row have been assigned
        valid values.
        "
    ::= { aggrMOEntry 6}

--
-- aggrDataTable: The Table of Data.  Each row represents a Data
```

```
--          set.  aggrCtlEntryID is the key to the table.
--          It is used to identify instances of the
--          aggregated MO that are present in the table.
--
aggrDataTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF AggrDataEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "Each row of this table contains information
         about an aggregateMO indexed by aggrCtlEntryID."
    ::= { aggrMIB 3 }

aggrDataEntry OBJECT-TYPE
    SYNTAX  AggrDataEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "Entry containing information pertaining to
         an aggregate MO."
    INDEX { aggrCtlEntryID }
    ::= { aggrDataTable 1 }

AggrDataEntry ::= SEQUENCE {
    aggrDataRecord
        AggrMOValue,
    aggrDataRecordCompressed
        AggrMOCCompressedValue,
    aggrDataErrorRecord
        AggrMOErrorStatus
}

aggrDataRecord OBJECT-TYPE
    SYNTAX  AggrMOValue
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The snapshot value of the aggregated MO.
         Note that the access privileges to this object will be
         governed by the access privileges of the component
         objects.  Thus, an entity attempting to access an
         instance of this MO MUST have access rights to all the
         component instance objects and this MO instance.
         "
    ::= { aggrDataEntry 1 }

aggrDataRecordCompressed OBJECT-TYPE
    SYNTAX  AggrMOCCompressedValue
```

```

MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The compressed value of the aggregated MO.
    The compression algorithm will depend on the
    aggrCtlCompressionAlgorithm given in the corresponding
    aggrCtlEntry.  If the value of the corresponding
    aggrCtlCompressionAlgorithm is (1) 'none', then the value
    of all instances of this object will be a string of zero
    length.
    Note that the access privileges to this object will be
    governed by the access privileges of the component
    objects.  Thus, an entity attempting to access an instance
    of this MO MUST have access rights to all the component
    instance objects and this MO instance.
    "
 ::= { aggrDataEntry 2}

aggrDataErrorRecord OBJECT-TYPE
    SYNTAX AggrMOErrorStatus
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The error status corresponding to the MO instances
        aggregated in aggrDataRecord (and
        aggrDataRecordCompressed). "
    ::= { aggrDataEntry 3}

-- Conformance information
aggrConformance OBJECT IDENTIFIER ::= { aggrMIB 4 }
aggrGroups OBJECT IDENTIFIER ::= { aggrConformance 1 }
aggrCompliances OBJECT IDENTIFIER ::= { aggrConformance 2 }

-- Compliance statements
aggrMibCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "The compliance statement for SNMP entities
        that implement the AGGREGATE-MIB."
    MODULE -- this module
        MANDATORY-GROUPS { aggrMibBasicGroup }
    ::= { aggrCompliances 1 }

-- Units of conformance
aggrMibBasicGroup OBJECT-GROUP
    OBJECTS {
        aggrCtlMOIndex,
        aggrCtlMODescr,

```



```

        aggrCtlCompressionAlgorithm,
        aggrCtlEntryOwner,
        aggrCtlEntryStorageType,
        aggrCtlEntryStatus,
        aggrMOInstance,
        aggrMODescr,
        aggrMOEntryStorageType,
        aggrMOEntryStatus,
        aggrDataRecord,
        aggrDataRecordCompressed,
        aggrDataErrorRecord
    }
    STATUS current
    DESCRIPTION
        "A collection of objects for aggregation of MOs."
    ::= { aggrGroups 1 }
END

```

```
TIME-AGGREGATE-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```

    MODULE-IDENTITY, experimental,
    OBJECT-TYPE, Opaque, Integer32
        FROM SNMPv2-SMI
    OwnerString
        FROM RMON-MIB
    RowStatus, StorageType, TEXTUAL-CONVENTION
        FROM SNMPv2-TC
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF
    SnmpAdminString
        FROM SNMP-FRAMEWORK-MIB;

```

```
tAggrMIB MODULE-IDENTITY
```

```

    LAST-UPDATED "200604270000Z"          -- 27 April 2006
    ORGANIZATION "Cyber Solutions Inc. NetMan Working Group"
    CONTACT-INFO

```

```

    "
        Glenn Mansfield Keeni
        Postal: Cyber Solutions Inc.
              6-6-3, Minami Yoshinari
              Aoba-ku, Sendai, Japan 989-3204.
        Tel: +81-22-303-4012
        Fax: +81-22-303-4015
        E-mail: glenn@cysols.com

```

```
    Support Group E-mail: mibsupport@cysols.com"
```

```
DESCRIPTION
```

"The MIB for servicing Time-Based aggregate objects.

Copyright (C) The Internet Society (2006). This version of this MIB module is part of RFC 4498; see the RFC itself for full legal notices.

"

REVISION "200604270000Z" -- 27th April, 2006
 DESCRIPTION "Initial version, published as RFC 4498."
 ::= { experimental 124 }

TAgrgrMOErrorStatus ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"This data type is used to model the error status of the sampled MO instance. The error status for a sampled MO instance is given in terms of two elements:

- o The moIndex, which indicates the sample number of the MO instance (starting at 1) in the value of the time-aggregated MO instance.
- o The moError, which indicates the error that was encountered in sampling that MO instance.

The syntax in ASN.1 Notation will be

```
ErrorStatus ::= SEQUENCE {
    moIndex Integer32,
    moError SnmpPduErrorStatus
}
```

```
TAgrgrMOErrorStatus ::= SEQUENCE OF {
    ErrorStatus
}
```

Note1: The command responder will supply values for all the samples of the MO instance. If an error is encountered for a sample, then the corresponding value will have an ASN.1 value NULL, and an error will be flagged in the corresponding TAgrgrMOErrorStatus object.

Only MOs for which errors have been encountered will the corresponding moIndex and moError values be set.

Note2: The error code for the component MO instances will be in accordance with the SnmpPduErrorStatus TC defined in the DISMAN-SCHEDULE-MIB[RFC3231].

"

SYNTAX Opaque (SIZE (0..1024))

TimeAggrMOValue ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"This data type is used to model the time-aggregated MOs. It

will be a sequence of values. The syntax in ASN.1 Notation will be

```
MOSampleValue ::= SEQUENCE {
    value ObjectSyntax
}
TimeAggrMOValue ::= SEQUENCE OF {
    MOSampleValue
}
```

where the first MOSampleValue, if any, will always be the timestamp of the first sample in the aggregated object. The subsequent values are the values of the MO instance sampled at the specified intervals for the specified number of times.

Note: The command generator will need to know the constituent MO instance and the sampling interval to correctly interpret TimeAggrMOValue.

```
"
SYNTAX      Opaque (SIZE (0..1024))
```

```
CompressedTimeAggrMOValue ::= TEXTUAL-CONVENTION
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"This data type is used to model the compressed
   TAgMOs."
```

```
SYNTAX      Opaque (SIZE (0..1024))
```

```
--
```

```
-- The Time-Based aggregation control table
```

```
--
```

```
tAggrCtlTable OBJECT-TYPE
```

```
SYNTAX      SEQUENCE OF TAggrCtlEntry
```

```
MAX-ACCESS  not-accessible
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"The Time-Based aggregation control table. It controls
   the aggregation of the samples of MO instances. There
   will be a row for each TAgMO.
```

```
"
```

```
::= { tAggrMIB 1 }
```

```
tAggrCtlEntry OBJECT-TYPE
```

```
SYNTAX      TAggrCtlEntry
```

```
MAX-ACCESS  not-accessible
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"A row of the control table that defines one Time-Based
   aggregate MO (TAgMO)."
```

```
INDEX { tAggrCtlEntryID }
```

```
::= { tAggrCtlTable 1 }
```

```
TAgrCtlEntry ::= SEQUENCE {
    tAggrCtlEntryID
        SnmpAdminString,
    tAggrCtlMOInstance
        OBJECT IDENTIFIER,
    tAggrCtlAgMODescr
        SnmpAdminString,
    tAggrCtlInterval
        Integer32,
    tAggrCtlSamples
        Integer32,
    tAggrCtlCompressionAlgorithm
        INTEGER,
    tAggrCtlEntryOwner
        OwnerString,
    tAggrCtlEntryStorageType
        StorageType,
    tAggrCtlEntryStatus
        RowStatus
}

tAggrCtlEntryID OBJECT-TYPE
    SYNTAX SnmpAdminString (SIZE(1..32))
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "A locally unique, administratively assigned name
        for this aggregated MO. It is used as an index to
        uniquely identify this row in the table."
    ::= { tAggrCtlEntry 1 }

tAggrCtlMOInstance OBJECT-TYPE
    SYNTAX OBJECT IDENTIFIER
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The sampled values of this MO instance will be
        aggregated by the TAgMO.
        "
    ::= { tAggrCtlEntry 2 }

tAggrCtlAgMODescr OBJECT-TYPE
    SYNTAX SnmpAdminString (SIZE(0..64))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "A textual description of the aggregate object."
    ::= { tAggrCtlEntry 3 }
```

```

tAggrCtlInterval OBJECT-TYPE
    SYNTAX Integer32
    UNITS "micro seconds"
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The interval, in microseconds, at which the MO instance
        pointed at by tAggrInstance will be sampled for
        Time-Based aggregation."
    ::= {tAggrCtlEntry 4}

tAggrCtlSamples OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The number of times at which the MO instance referred
        to by tAggrInstance will be sampled for Time-Based
        aggregation."
    ::= {tAggrCtlEntry 5}

-- only one compression algorithm is defined as of now.
tAggrCtlCompressionAlgorithm OBJECT-TYPE
    SYNTAX INTEGER {
        none      (1),
        deflate   (2)
    }
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The compression algorithm that will be used by
        the agent to compress the value of the TAgMO.
        The deflate algorithm and corresponding data format
        specification is described in RFC 1951. It is
        compatible with the widely used gzip utility."
    ::= {tAggrCtlEntry 6}

REFERENCE
    "RFC1951 : DEFLATE Compressed Data Format Specification
    version 1.3"

DEFVAL { none }
::= {tAggrCtlEntry 6}

tAggrCtlEntryOwner OBJECT-TYPE
    SYNTAX OwnerString
    MAX-ACCESS read-create
    STATUS current

```

DESCRIPTION

"A textual description of the entity that created this entry.

"

::= {tAggrCtlEntry 7}

tAggrCtlEntryStorageType OBJECT-TYPE

SYNTAX StorageType

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object defines whether the parameters defined in this row are kept in volatile storage and lost upon reboot or backed up by non-volatile (permanent) storage.

Conceptual rows having the value 'permanent' need not allow write-access to any columnar objects in the row.

"

::= {tAggrCtlEntry 8}

tAggrCtlEntryStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The row status variable, used according to row installation and removal conventions.

Objects in a row can be modified only when the value of this object in the corresponding conceptual row is not 'active'.

Thus, to modify one or more of the objects in this conceptual row,

- a. change the row status to 'notInService',
- b. change the values of the row, and
- c. change the row status to 'active'.

The tAggrCtlEntryStatus may be changed to 'active' iff all the MOs in the conceptual row have been assigned valid values.

"

::= {tAggrCtlEntry 9}

--

-- tAggrDataTable: The data table.

--

tAggrDataTable OBJECT-TYPE

SYNTAX SEQUENCE OF TAggrDataEntry

```

MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "This is the data table. Each row of this table contains
    information about a TAgMO indexed by tAggrCtlEntryID.
    tAggrCtlEntryID is the key to the table. It is used to
    identify instances of the TAgMO that are present in the
    table."
    ::= {tAggrMIB 2}

tAggrDataEntry OBJECT-TYPE
    SYNTAX TAggrDataEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Entry containing information pertaining
        to a TAgMO."
    INDEX {tAggrCtlEntryID}
    ::= {tAggrDataTable 1}

TAggrDataEntry ::= SEQUENCE {
    tAggrDataRecord
        TimeAggrMOValue,
    tAggrDataRecordCompressed
        CompressedTimeAggrMOValue,
    tAggrDataErrorRecord
        TAggrMOErrorStatus
}

tAggrDataRecord OBJECT-TYPE
    SYNTAX TimeAggrMOValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The snapshot value of the TAgMO."
    ::= { tAggrDataEntry 1}

tAggrDataRecordCompressed OBJECT-TYPE
    SYNTAX CompressedTimeAggrMOValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The compressed value of the TAgMO.
        The compression algorithm will depend on the
        tAggrCtlCompressionAlgorithm given in the corresponding
        tAggrCtlEntry. If the value of the corresponding
        tAggrCtlCompressionAlgorithm is (1) 'none', then the

```

value of all instances of this object will be a string of zero length.
 Note that the access privileges to this object will be governed by the access privileges of the corresponding MO instance. Thus, an entity attempting to access an instance of this MO MUST have access rights to the instance object pointed at by tAggrCtlMOInstance and this MO instance.

```

"
 ::= { tAggrDataEntry 2}

tAggrDataErrorRecord OBJECT-TYPE
    SYNTAX TAggrMOErrorStatus
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The error status corresponding to the MO instance
        samples aggregated in tAggrDataRecord (and
        tAggrDataRecordCompressed)."
```

```

 ::= { tAggrDataEntry 3}

-- Conformance information
tAggrConformance OBJECT IDENTIFIER ::= { tAggrMIB 3 }
tAggrGroups      OBJECT IDENTIFIER ::= { tAggrConformance 1 }
tAggrCompliances OBJECT IDENTIFIER ::= { tAggrConformance 2 }

-- Compliance statements
tAggrMibCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "The compliance statement for SNMP entities
        that implement the TIME-AGGREGATE-MIB."
    MODULE -- this module
        MANDATORY-GROUPS { tAggrMibBasicGroup }
    ::= { tAggrCompliances 1 }

-- Units of conformance
tAggrMibBasicGroup OBJECT-GROUP
    OBJECTS {
        tAggrCtlMOInstance,
        tAggrCtlAgMODescr,
        tAggrCtlInterval,
        tAggrCtlSamples,
        tAggrCtlCompressionAlgorithm,
        tAggrCtlEntryOwner,
        tAggrCtlEntryStorageType,
        tAggrCtlEntryStatus,
```



```

        tAggrDataRecord,
        tAggrDataRecordCompressed,
        tAggrDataErrorRecord
    }
    STATUS    current
    DESCRIPTION
        "A collection of objects for Time-Based aggregation
        of MOs."
    ::= { tAggrGroups 1 }
END

```

7. Security Considerations

There are management objects in the MIB modules defined in this document that have a MAX-ACCESS clause of read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations. The objects and corresponding vulnerabilities are discussed below.

The following MOs are used to configure an agent that implements the aggregate MIB modules.

```

aggrCtlMOIndex,
aggrCtlMODescr,
aggrCtlCompressionAlgorithm,
aggrCtlEntryOwner,
aggrCtlEntryStorageType,
aggrCtlEntryStatus,
aggrMOInstance,
aggrMODescr,
aggrMOEntryStorageType,
aggrMOEntryStatus,
tAggrCtlMOInstance,
tAggrCtlAgMODescr,
tAggrCtlInterval,
tAggrCtlSamples,
tAggrCtlCompressionAlgorithm,
tAggrCtlEntryOwner,
tAggrCtlEntryStorageType,
tAggrCtlEntryStatus,

```

Access to these objects may be abused to affect the operation of the data collection system. In particular,

- by changing the value of an instance of `aggrCtlEntryStatus`, `tAggrCtlEntryStatus`, `aggrMOEntryStatus`, or `tAggrMOEntryStatus` to 'notInService' or 'destroy', the data aggregation operation for the corresponding entry will become unavailable to the management system.
- by changing the value of an instance of `aggrMOInstance` or `tAggrCtlMOInstance`, the data aggregation operation may be subverted. This may result in wrong information being fed to the management system.
- by adding several rows in the `aggrMOTable` corresponding to an aggregate MO, it is possible to make the value of the aggregate MOs very large. A similar effect may be achieved by manipulating the value of the `tAggrCtlSamples` instance corresponding to a Time-Based aggregate MO. This could result in very heavy management traffic and/or fragmentation of response packets. In some cases the responder may refuse to send the data and will simply respond with an error message indicating that the response packet size is too big.

An entity attempting to access an instance of an aggregated MO MUST have access rights to all the component instance objects and the aggregate MO instance. An implementation MUST follow this requirement. Lax adherence to this requirement will breach the security model and make the system vulnerable to illegal accesses.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

8. IANA Considerations

The MIB modules in this document use the following IANA-assigned OBJECT IDENTIFIER values, recorded in the SMI Numbers registry:

Descriptor	OBJECT IDENTIFIER value
-----	-----
aggrMIB	{ experimental 123 }
tAggrMIB	{ experimental 124 }

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2578] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC2579] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.
- [RFC2819] Waldbusser, S., "Remote Network Monitoring Management Information Base", STD 59, RFC 2819, May 2000.
- [RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.
- [RFC3231] Levi, D. and J. Schoenwaelder, "Definitions of Managed Objects for Scheduling Management Operations", RFC 3231, January 2002.
- [RFC1951] Deutsch, P., "DEFLATE Compressed Data Format Specification version 1.3", RFC 1951, May 1996.

9.2. Informative References

- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

10. Acknowledgements

This document is the product of discussions and deliberations carried out in the WIDE-netman group. Bert Wijnen and Glenn Waters reviewed the document and provided valuable comments.

Authors' Addresses

Glenn Mansfield Keeni
Cyber Solutions Inc.
6-6-3 Minami Yoshinari
Aoba-ku, Sendai 989-3204
Japan

Phone: +81-22-303-4012
EMail: glenn@cysols.com

Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in BCP 78 and at www.rfc-editor.org/copyright.html, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

