

Definitions of Managed Objects for IEEE 802.12 Repeater Devices

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.

Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in TCP/IP-based internets. In particular, it defines objects for managing network repeaters based on IEEE 802.12.

Table of Contents

1. The SNMP Network Management Framework	2
1.1. Object Definitions	2
2. Overview	2
2.1. Repeater Management Model	3
2.2. MAC Addresses	4
2.3. Master Mode and Slave Mode	4
2.4. IEEE 802.12 Training Frames	4
2.5. Structure of the MIB	6
2.5.1. Basic Definitions	7
2.5.2. Monitor Definitions	7
2.5.3. Address Tracking Definitions	7
2.6. Relationship to other MIBs	7
2.6.1. Relationship to MIB-II	7
2.6.1.1. Relationship to the 'system' group	7
2.6.1.2. Relationship to the 'interfaces' group	8
2.6.2. Relationship to the 802.3 Repeater MIB	8

2.7. Mapping of IEEE 802.12 Managed Objects	9
3. Definitions	12
4. Acknowledgements	53
5. References	53
6. Security Considerations	54
7. Author's Address	55
8. Full Copyright Statement	56

1. The SNMP Network Management Framework

The SNMP Network Management Framework consists of several components. For the purpose of this specification, the applicable components of the Framework are the SMI and related documents [2, 3, 4], which define the mechanisms used for describing and naming objects for the purpose of management.

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

1.1. Object Definitions

Managed objects are accessed via a virtual information store, termed the Management Information Base (MIB). Objects in the MIB are defined using the subset of Abstract Syntax Notation One (ASN.1) [1] defined in the SMI [2]. 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.

2. Overview

Instances of these object types represent attributes of an IEEE 802.12 repeater, as defined by Section 12, "RMAC Protocol" in IEEE Standard 802.12-1995 [6].

The definitions presented here are based on Section 13, "Layer management functions and services", and Annex C, "GDMO Specifications for Demand Priority Managed Objects" of IEEE Standard 802.12-1995 [6].

Implementors of these MIB objects should note that the IEEE document explicitly describes (in the form of Pascal pseudocode) when, where, and how various repeater attributes are measured. The IEEE document also describes the effects of repeater actions that may be invoked by manipulating instances of the MIB objects defined here.

The counters in this document are defined to be the same as those counters in IEEE Standard 802.12-1995, with the intention that the same instrumentation can be used to implement both the IEEE and IETF management standards.

2.1. Repeater Management Model

The model used in the design of this MIB allows for a managed system to contain one or more managed 802.12 repeaters, and one or more managed 802.12 repeater ports.

A repeater port may be thought of as a source of traffic into a repeater in the system. The `vgRptrBasicPortTable` contains entries for each physical repeater port in the managed system. An implementor may choose to separate these ports into "groups". For example, a group may be used to represent a field-replaceable unit, so that the port numbering may match the numbering in the hardware implementation. Note that this group mapping is recommended but optional. An implementor may choose to put all of the system's ports into a single group, or to divide the ports into groups that do not match physical divisions. Each group within the system is uniquely identified by a group number. Each port within a system is uniquely identified by a combination of group number and port number. The method of numbering groups and ports is implementation-specific. Both groups and ports may be sparsely numbered.

In addition to the externally visible ports, some implementations may have internal ports that are not obvious to the end-user but are nevertheless sources of traffic into the repeater system. Examples include internal management ports, through which an agent communicates, and ports connecting to a backplane internal to the implementation. It is the decision of the implementor to select the appropriate group(s) in which to place internal ports.

Managed repeaters in the system are represented by entries in the `vgRptrInfoTable`. There may be multiple repeaters in the managed system. They are uniquely identified by a repeater number. The method of numbering repeaters is implementation-specific. Each port will either be associated with one of the repeaters, or isolated (a so-called "trivial" repeater). The set of ports associated with a single repeater will be in the same contention domain, and will be participating in the same instance of the Demand Priority Access Method protocol. The mapping of ports to repeaters may be static or dynamic. A column in the `vgRptrBasicPortTable`, `vgRptrPortRptrInfoIndex`, indicates the repeater that the port is currently associated with. The method for assigning a port to a repeater is implementation-specific.

2.2. MAC Addresses

All representations of MAC addresses in this MIB module are in "canonical" order defined by 802.1a, i.e., as if it were transmitted least significant bit first. This is true even if the repeater is operating in token ring framing mode, which requires MAC addresses to be transmitted most significant bit first.

2.3. Master Mode and Slave Mode

In an IEEE 802.12 network, "master" devices act as network controllers to decide when to grant requesting end-nodes permission to transmit. These master devices may be repeaters, or other active controller devices such as switches.

Devices which do not act as network controllers, such as end-nodes or passive switches, are considered to be operating in "slave" mode.

An 802.12 repeater always acts in "master" mode on its local ports, which may connect to end nodes, switch or other device ports acting in "slave" mode, or lower-level repeaters in a cascade. It acts in "slave" mode on cascade ports, which may connect to an upper-level repeater in a cascade, or to switch or other device ports operating in "master" mode.

2.4. IEEE 802.12 Training Frames

Training frames are special MAC frames that are used only during link initialization. Training frames are initially constructed by the device at the "lower" end of a link, which is the slave mode device for the link. The training frame format is as follows:

```
+---+---+-----+-----+-----+---+
| DA | SA | Req Config | Allow Config | Data | FCS |
+---+---+-----+-----+-----+---+
```

DA = destination address (six octets)

SA = source address (six octets)

Req Config = requested configuration (2 octets)

Allow Config = allowed configuration (2 octets)

Data = data (594 to 675 octets)

FCS = frame check sequence (4 octets)

Training frames are always sent with a null destination address. To pass training, an end node must use its source address in the source address field of the training frame. A repeater may use a non-null source address if it has one, or it may use a null source address.

The requested configuration field allows the slave mode device to inform the master mode device about itself and to request configuration options. The training response frame from the master mode device contains the slave mode device's requested configuration from the training request frame. The currently defined format of the requested configuration field as defined in the IEEE Standard 802.12-1995 standard is shown below. Please refer to the most current version of the IEEE document for a more up to date description of this field. In particular, the reserved bits may be used in later versions of the standard.

First Octet:	Second Octet:
7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
+-+-+-----+	+-+-+-----+
v v v r r r r	r r r F F P P R
+-+-+-----+	+-+-+-----+

vvv: The version of the 802.12 training protocol with which the training initiator is compliant. The current version is 100. Note that because of the different bit ordering used in IEEE and IETF documents, this value corresponds to version 1.

r: Reserved bits (set to zero)

FF: 00 = frameType88023

01 = frameType88025

10 = reserved

11 = frameTypeEither

PP: 00 = singleAddressMode

01 = promiscuousMode

10 = reserved

11 = reserved

R: 0 = the training initiator is an end node

1 = the training initiator is a repeater

The allowed configuration field allows the master mode device to respond with the allowed configuration. The slave mode device sets the contents of this field to all zero bits. The master mode device sets the allowed configuration field as follows:

First Octet:	Second Octet:
7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
+-+-+-----+	+-+-+-----+
v v v D C N r r	r r r F F P P R
+-+-+-----+	+-+-+-----+

vvv: The version of the 802.12 training protocol with which the training responder is compliant. The current version is 100. Note that because of the different bit ordering used in IEEE and IETF documents, this value corresponds to version 1.

D: 0 = No duplicate address has been detected.
1 = Duplicate address has been detected.

C: 0 = The requested configuration is compatible with the network and the attached port.
1 = The requested configuration is not compatible with the network and/or the attached port. In this case, the FF, PP, and R bits indicate a configuration that would be allowed.

N: 0 = Access will be allowed, providing the configuration is compatible (C = 0).
1 = Access is not granted because of security restrictions.

r: Reserved bits (set to zero).

FF: 00 = frameType88023 will be used.
01 = frameType88025 will be used.
10 = reserved
11 = reserved

PP: 00 = singleAddressMode
01 = promiscuousMode
10 = reserved
11 = reserved

R: 0 = Requested access as an end node is allowed.
1 = Requested access as a repeater is allowed.

Again, note that the most recent version of the IEEE 802.12 standard should be consulted for the most up to date definition of the requested configuration and allowed configuration fields.

The data field contains between 594 and 675 octets and is filled in by the training initiator. The first 55 octets may be used for vendor specific protocol information. The remaining octets are all zeros. The length of the training frame combined with the requirement that 24 consecutive training frames be exchanged without error to complete training ensures that marginal links will not complete training.

2.5. Structure of the MIB

Objects in this MIB are arranged into OID subtrees, each of which contains a set of related objects within a broad functional category. These subtrees are intended for organizational convenience ONLY, and have no relation to the conformance groups defined later in the document.

2.5.1. Basic Definitions

The basic definitions include objects for managing the basic status and control parameters for each repeater within the managed system, for the port groups within the managed system, and for the individual ports themselves.

2.5.2. Monitor Definitions

The monitor definitions include monitoring statistics for each repeater within the system and for individual ports.

2.5.3. Address Tracking Definitions

This collection includes objects for tracking the MAC addresses of the DTEs attached to the ports within the system.

Note that this MIB also includes by reference a collection of objects from the 802.3 Repeater MIB which may be used for mapping the topology of a network. These definitions are based on a technology which has been patented by Hewlett-Packard Company (HP). HP has granted rights to this technology to implementors of this MIB. See [8] and [9] for details.

2.6. Relationship to other MIBs

2.6.1. Relationship to MIB-II

It is assumed that a repeater implementing this MIB will also implement (at least) the 'system' group defined in MIB-II [5].

2.6.1.1. Relationship to the 'system' group

In MIB-II, the 'system' group is defined as being mandatory for all systems such that each managed entity contains one instance of each object in the 'system' group. Thus, those objects apply to the entity even if the entity's sole functionality is management of repeaters.

Note that all of the managed repeaters (i.e. entries in the `vgRptrInfoTable`) will normally exist within a single naming scope. Therefore, there will normally only be a single instance of each of the objects in the system group for the entire managed repeater system regardless of how many managed repeaters there are in the system.

2.6.1.2. Relationship to the 'interfaces' group

In MIB-II, the 'interfaces' group is defined as being mandatory for all systems and contains information on an entity's interfaces, where each interface is thought of as being attached to a 'subnetwork'. (Note that this term is not to be confused with 'subnet' which refers to an addressing partitioning scheme used in the Internet suite of protocols.)

This Repeater MIB uses the notion of ports on a repeater. The concept of a MIB-II interface has NO specific relationship to a repeater's port. Therefore, the 'interfaces' group applies only to the one (or more) network interfaces on which the entity managing the repeater sends and receives management protocol operations, and does not apply to the repeater's ports.

This is consistent with the physical-layer nature of a repeater. An 802.12 repeater has an RMAC implementation, which acts as the repeater end of the Demand Priority Access Method, but does not contain a DTE MAC implementation, and does not pass packets up to higher-level protocol entities for processing.

(When a network management entity is observing a repeater, it may appear as though the repeater is passing packets to a higher-level protocol entity. However, this is only a means of implementing management, and this passing of management information is not part of the repeater functionality.)

2.6.2. Relationship to the 802.3 Repeater MIB

An IEEE 802.12 repeater can be configured to operate in either ethernet or token ring framing mode. This only affects the frame format and address bit order of the frames on the wire. An 802.12 network does not use the media access protocol for either ethernet or token ring. Instead, IEEE 802.12 defines its own media access protocol, the Demand Priority Access Method (DPAM).

There is an existing standards-track MIB module for instrumenting IEEE 802.3 repeaters [7]. That MIB module is designed to instrument the operation of the repeater in a network implementing the 802.3 media access protocol. Therefore, much of that MIB does not apply to 802.12 repeaters.

However, the 802.3 Repeater MIB also contains a collection of objects that may be used to map the topology of a network. These objects are contained in a separable OBJECT-GROUP, are not 802.3-specific, and are considered useful for 802.12 repeaters. In addition, the layer

management clause of the IEEE 802.12 specification includes similar functionality. Therefore, vendors of agents for 802.12 repeaters are encouraged to implement the snmpRptrGrpRptrAddrSearch OBJECT-GROUP defined in the 802.3 Repeater MIB.

2.7. Mapping of IEEE 802.12 Managed Objects

IEEE 802.12 Managed Object	Corresponding SNMP Object
oRepeater	
.aCurrentFramingType	vgRptrInfoCurrentFramingType
.aDesiredFramingType	vgRptrInfoDesiredFramingType
.aFramingCapability	vgRptrInfoFramingCapability
.aMACAddress	vgRptrInfoMACAddress
.aRepeaterHealthState	vgRptrInfoOperStatus
.aRepeaterID	vgRptrInfoIndex
.aRepeaterSearchAddress	SNMP-REPEATER-MIB - rptrAddrSearchAddress
.aRepeaterSearchGroup	SNMP-REPEATER-MIB - rptrAddrSearchGroup
.aRepeaterSearchPort	SNMP-REPEATER-MIB - rptrAddrSearchPort
.aRepeaterSearchState	SNMP-REPEATER-MIB - rptrAddrSearchState
.aRMACVersion	vgRptrInfoTrainingVersion
.acRepeaterSearchAddress	SNMP-REPEATER-MIB - rptrAddrSearchAddress
.acResetRepeater	vgRptrInfoReset
.nRepeaterHealth	vgRptrHealth
.nRepeaterReset	vgRptrResetEvent
oGroup	
.aGroupCablesBundled	vgRptrGroupCablesBundled
.aGroupID	vgRptrGroupIndex
.aGroupPortCapacity	vgRptrGroupPortCapacity
oPort	
.aAllowableTrainingType	vgRptrPortAllowedTrainType
.aBroadcastFramesReceived	vgRptrPortBroadcastFrames
.aCentralMgmtDetectedDupAddr	vgRptrMgrDetectedDupAddress
.aDataErrorFramesReceived	vgRptrPortDataErrorFrames
.aHighPriorityFramesReceived	vgRptrPortHighPriorityFrames
.aHighPriorityOctetsReceived	vgRptrPortHCHighPriorityOctets, or vgRptrPortHighPriorityOctets and vgRptrPortHighPriOctetRollovers
.aIPMFramesReceived	vgRptrPortIPMFrames
.aLastTrainedAddress	vgRptrAddrLastTrainedAddress
.aLastTrainingConfig	vgRptrPortLastTrainConfig

.aLocalRptrDetectedDupAddr	vgRptrRptrDetectedDupAddress
.aMulticastFramesReceived	vgRptrPortMulticastFrames
.aNormalPriorityFramesReceived	vgRptrPortNormPriorityFrames
.aNormalPriorityOctetsReceived	vgRptrPortHCNormPriorityOctets, or vgRptrPortNormPriorityOctets and vgRptrPortNormPriOctetRollovers
.aNullAddressedFramesReceived	vgRptrPortNullAddressedFrames
.aOctetsInUnreadableFramesRcvd	vgRptrPortHCUnreadableOctets, or vgRptrPortUnreadableOctets and vgRptrPortUnreadOctetRollovers
.aOversizeFramesReceived	vgRptrPortOversizeFrames
.aPortAdministrativeState	vgRptrPortAdminStatus
.aPortID	vgRptrPortIndex
.aPortStatus	vgRptrPortOperStatus
.aPortType	vgRptrPortType
.aPriorityEnable	vgRptrPortPriorityEnable
.aPriorityPromotions	vgRptrPortPriorityPromotions
.aReadableFramesReceived	vgRptrPortReadableFrames
.aReadableOctetsReceived	vgRptrPortHCReadableOctets, or vgRptrPortReadableOctets and vgRptrPortReadOctetRollovers
.aSupportedCascadeMode	vgRptrPortSupportedCascadeMode
.aSupportedPromiscMode	vgRptrPortSupportedPromiscMode
.aTrainedAddressChanges	vgRptrAddrTrainedAddressChanges
.aTrainingResult	vgRptrPortTrainingResult
.aTransitionsIntoTraining	vgRptrPortTransitionToTrainings
.acPortAdministrativeControl	vgRptrPortAdminStatus

The following IEEE 802.12 managed objects have not been included in the 802.12 Repeater MIB for the indicated reasons.

IEEE 802.12 Managed Object	Disposition
oRepeater	
.aGroupMap	Can be determined by GetNext sweep of vgRptrBasicGroupTable
.aRepeaterGroupCapacity	Meaning is unclear in many repeater implementations. For example, some cards may have daughter cards which make group capacity change depending on the cards installed. Meaning is also unclear in a stackable implementation. Also, since groups are not required to be numbered from 1..capacity, but may be computed algorithmically or

	related to Entity MIB indices, this object was not considered useful.
.aRepeaterHealthData	Since the data is implementation specific and non-interoperable, it was not considered useful.
.aRepeaterHealthText	Implementation experience with similar object in 802.3 Rptr MIB indicated it was not useful.
.acExecuteNonDisruptiveSelfTest	Implementation experience with similar object in 802.3 Rptr MIB indicated it was not useful.
.nGroupMapChange	Since aGroupMap was not included, a notification of a change in that object was not needed.
oGroup	
.aPortMap	Can be determined by GetNext sweep of vgRptrBasicPortTable
.nPortMapChange	Since aPortMap was not included, a notification of a change in that object was not needed.
oPort	
.aMediaType	This object is a function of the Physical Media Dependent (PMD) layer, which is defined differently for each type of network. For an 802.3 network, .aMediaType corresponds to the PMD definitions in the 802.3 MAU MIB. For management of an 802.12 network, mapping of this object is deferred to future work on an 802.12 PMD MIB which will include both repeater and interface PMD information and redundant link support.

3. Definitions

```
DOT12-RPTR-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    mib-2, Integer32, Counter32, Counter64,
    OBJECT-TYPE, MODULE-IDENTITY, NOTIFICATION-TYPE
    FROM SNMPv2-SMI
    MacAddress, TruthValue, TimeStamp
    FROM SNMPv2-TC
    MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
    FROM SNMPv2-CONF;
```

```
vgRptrMIB MODULE-IDENTITY
```

```
    LAST-UPDATED "9705192256Z" -- May 19, 1997
    ORGANIZATION "IETF 100VG-AnyLAN Working Group"
    CONTACT-INFO
        "WG E-mail: vgmib@hprnd.rose.hp.com
```

```
        Chair: Jeff Johnson
        Postal: RedBack Networks
                2570 North First Street, Suite 410
                San Jose, CA 95131
        Tel: +1 408 571 2699
        Fax: +1 408 571 2698
        E-mail: jeff@redbacknetworks.com
```

```
        Editor: John Flick
        Postal: Hewlett Packard Company
                8000 Foothills Blvd. M/S 5556
                Roseville, CA 95747-5556
        Tel: +1 916 785 4018
        Fax: +1 916 785 3583
        E-mail: johnf@hprnd.rose.hp.com"
```

```
DESCRIPTION
```

```
    "This MIB module describes objects for managing
    IEEE 802.12 repeaters."
```

```
 ::= { mib-2 53 }
```

```
vgRptrObjects      OBJECT IDENTIFIER ::= { vgRptrMIB 1 }
vgRptrBasic        OBJECT IDENTIFIER ::= { vgRptrObjects 1 }
vgRptrBasicRptr    OBJECT IDENTIFIER ::= { vgRptrBasic 1 }
```

```
vgRptrInfoTable OBJECT-TYPE
```

```
    SYNTAX      SEQUENCE OF VgRptrInfoEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
```

"A table of information about each 802.12 repeater in the managed system."
 ::= { vgRptrBasicRptr 1 }

vgRptrInfoEntry OBJECT-TYPE
 SYNTAX VgRptrInfoEntry
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION
 "An entry in the table, containing information about a single repeater."
 INDEX { vgRptrInfoIndex }
 ::= { vgRptrInfoTable 1 }

VgRptrInfoEntry ::=

SEQUENCE {	
vgRptrInfoIndex	Integer32,
vgRptrInfoMACAddress	MacAddress,
vgRptrInfoCurrentFramingType	INTEGER,
vgRptrInfoDesiredFramingType	INTEGER,
vgRptrInfoFramingCapability	INTEGER,
vgRptrInfoTrainingVersion	INTEGER,
vgRptrInfoOperStatus	INTEGER,
vgRptrInfoReset	INTEGER,
vgRptrInfoLastChange	TimeStamp
}	

vgRptrInfoIndex OBJECT-TYPE
 SYNTAX Integer32 (1..2147483647)
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION
 "A unique identifier for the repeater for which this entry contains information. The numbering scheme for repeaters is implementation specific."
 REFERENCE
 "IEEE Standard 802.12-1995, 13.2.4.2.1, aRepeaterID."
 ::= { vgRptrInfoEntry 1 }

vgRptrInfoMACAddress OBJECT-TYPE
 SYNTAX MacAddress
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "The MAC address used by the repeater when it initiates training on the uplink port. Repeaters are allowed to train with an assigned MAC address

or a null (all zeroes) MAC address."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.2.1,
aMACAddress."

::= { vgRptrInfoEntry 2 }

vgRptrInfoCurrentFramingType OBJECT-TYPE

SYNTAX INTEGER {
frameType88023(1),
frameType88025(2)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The type of framing (802.3 or 802.5) currently
in use by the repeater."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.2.1,
aCurrentFramingType."

::= { vgRptrInfoEntry 3 }

vgRptrInfoDesiredFramingType OBJECT-TYPE

SYNTAX INTEGER {
frameType88023(1),
frameType88025(2)
}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The type of framing which will be used by the
repeater after the next time it is reset."

The value of this object should be preserved
across repeater resets and power failures."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.2.1,
aDesiredFramingType."

::= { vgRptrInfoEntry 4 }

vgRptrInfoFramingCapability OBJECT-TYPE

SYNTAX INTEGER {
frameType88023(1),
frameType88025(2),
frameTypeEither(3)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The type of framing this repeater is capable of supporting."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.2.1, aFramingCapability."

::= { vgRptrInfoEntry 5 }

vgRptrInfoTrainingVersion OBJECT-TYPE

SYNTAX INTEGER (0..7)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The highest version bits (vvv bits) supported by the repeater during training."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.2.1, aRMACVersion."

::= { vgRptrInfoEntry 6 }

vgRptrInfoOperStatus OBJECT-TYPE

SYNTAX INTEGER {
 other(1),
 ok(2),
 generalFailure(3)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The vgRptrInfoOperStatus object indicates the operational state of the repeater."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.2.1, aRepeaterHealthState."

::= { vgRptrInfoEntry 7 }

vgRptrInfoReset OBJECT-TYPE

SYNTAX INTEGER {
 noReset(1),
 reset(2)
}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Setting this object to reset(2) causes the repeater to transition to its initial state as specified in clause 12 [IEEE Std 802.12]."

Setting this object to noReset(1) has no effect. The agent will always return the value noReset(1) when this object is read.

After receiving a request to set this variable to reset(2), the agent is allowed to delay the reset for a short period. For example, the implementor may choose to delay the reset long enough to allow the SNMP response to be transmitted. In any event, the SNMP response must be transmitted.

This action does not reset the management counters defined in this document nor does it affect the vgRptrPortAdminStatus parameters. Included in this action is the execution of a disruptive Self-Test with the following characteristics:

- 1) The nature of the tests is not specified.
- 2) The test resets the repeater but without affecting configurable management information about the repeater.
- 3) Packets received during the test may or may not be transferred.
- 4) The test does not interfere with management functions.

After performing this self-test, the agent will update the repeater health information (including vgRptrInfoOperStatus), and send a vgRptrResetEvent."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.2.2, acResetRepeater."

::= { vgRptrInfoEntry 8 }

vgRptrInfoLastChange OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of sysUpTime when any of the following conditions occurred:

- 1) agent cold- or warm-started;
- 2) this instance of repeater was created (such as when a device or module was added to the system);


```

        3) a change in the value of
            vgRptrInfoOperStatus;
        4) ports were added or removed as members of
            the repeater; or
        5) any of the counters associated with this
            repeater had a discontinuity."
 ::= { vgRptrInfoEntry 9 }

vgRptrBasicGroup OBJECT IDENTIFIER ::= { vgRptrBasic 2 }

vgRptrBasicGroupTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF VgRptrBasicGroupEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing information about groups of
         ports."
    ::= { vgRptrBasicGroup 1 }

vgRptrBasicGroupEntry OBJECT-TYPE
    SYNTAX      VgRptrBasicGroupEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry in the vgRptrBasicGroupTable, containing
         information about a single group of ports."
    INDEX       { vgRptrGroupIndex }
    ::= { vgRptrBasicGroupTable 1 }

VgRptrBasicGroupEntry ::=
    SEQUENCE {
        vgRptrGroupIndex                Integer32,
        vgRptrGroupObjectID              OBJECT IDENTIFIER,
        vgRptrGroupOperStatus             INTEGER,
        vgRptrGroupPortCapacity           Integer32,
        vgRptrGroupCablesBundled          INTEGER
    }

vgRptrGroupIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2146483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object identifies the group within the
         system for which this entry contains information.
         The numbering scheme for groups is implementation
         specific."
    REFERENCE

```

```

        "IEEE Standard 802.12-1995, 13.2.4.4.1,
        aGroupID."
 ::= { vgRptrBasicGroupEntry 1 }

vgRptrGroupObjectID OBJECT-TYPE
    SYNTAX      OBJECT IDENTIFIER
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The vendor's authoritative identification of the
        group. This value may be allocated within the
        SMI enterprises subtree (1.3.6.1.4.1) and
        provides a straight-forward and unambiguous means
        for determining what kind of group is being
        managed.

        For example, this object could take the value
        1.3.6.1.4.1.4242.1.2.14 if vendor 'Flintstones,
        Inc.' was assigned the subtree 1.3.6.1.4.1.4242,
        and had assigned the identifier
        1.3.6.1.4.1.4242.1.2.14 to its 'Wilma Flintstone
        6-Port Plug-in Module.'"
 ::= { vgRptrBasicGroupEntry 2 }

vgRptrGroupOperStatus OBJECT-TYPE
    SYNTAX      INTEGER {
        other(1),
        operational(2),
        malfunctioning(3),
        notPresent(4),
        underTest(5),
        resetInProgress(6)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "An object that indicates the operational status
        of the group.

        A status of notPresent(4) indicates that the
        group is temporarily or permanently physically
        and/or logically not a part of the system. It
        is an implementation-specific matter as to
        whether the agent effectively removes notPresent
        entries from the table.

        A status of operational(2) indicates that the
        group is functioning, and a status of

```

malfunctioning(3) indicates that the group is malfunctioning in some way."
 ::= { vgRptrBasicGroupEntry 3 }

vgRptrGroupPortCapacity OBJECT-TYPE

SYNTAX Integer32 (1..2146483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The vgRptrGroupPortCapacity is the number of ports that can be contained within the group. Valid range is 1-2147483647. Within each group, the ports are uniquely numbered in the range from 1 to vgRptrGroupPortCapacity.

Some ports may not be present in the system, in which case the actual number of ports present will be less than the value of vgRptrGroupPortCapacity. The number of ports present is never greater than the value of vgRptrGroupPortCapacity.

Note: In practice, this will generally be the number of ports on a module, card, or board, and the port numbers will correspond to numbers marked on the physical embodiment."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.4.1, aGroupPortCapacity."

::= { vgRptrBasicGroupEntry 4 }

vgRptrGroupCablesBundled OBJECT-TYPE

SYNTAX INTEGER {
 someCablesBundled(1),
 noCablesBundled(2)
 }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object is used to indicate whether there are any four-pair UTP links connected to this group that are contained in a cable bundle with multiple four-pair groups (e.g. a 25-pair bundle). Bundled cable may only be used for repeater-to-end node links where the end node is not in promiscuous mode.

When a broadcast or multicast packet is received from a port on this group that is not a

promiscuous or cascaded port, the packet will be buffered completely before being repeated if this object is set to 'someCablesBundled(1)'. When this object is equal to 'noCablesBundled(2)', all packets received from ports on this group will be repeated as the frame is being received.

Note that the value 'someCablesBundled(1)' will work in the vast majority of all installations, regardless of whether or not any cables are physically in a bundle, since packets received from promiscuous and cascaded ports automatically avoid the store and forward. The main situation in which 'noCablesBundled(2)' is beneficial is when there is a large amount of multicast traffic and the cables are not in a bundle.

The value of this object should be preserved across repeater resets and power failures."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.4.1, aGroupCablesBundled."

::= { vgRptrBasicGroupEntry 5 }

vgRptrBasicPort OBJECT IDENTIFIER ::= { vgRptrBasic 3 }

vgRptrBasicPortTable OBJECT-TYPE

SYNTAX SEQUENCE OF VgRptrBasicPortEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A table containing configuration and status information about 802.12 repeater ports in the system. The number of entries is independent of the number of repeaters in the managed system."

::= { vgRptrBasicPort 1 }

vgRptrBasicPortEntry OBJECT-TYPE

SYNTAX VgRptrBasicPortEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the vgRptrBasicPortTable, containing information about a single port."

INDEX { vgRptrGroupIndex, vgRptrPortIndex }

::= { vgRptrBasicPortTable 1 }

VgRptrBasicPortEntry ::=

```

SEQUENCE {
    vgRptrPortIndex                Integer32,
    vgRptrPortType                 INTEGER,
    vgRptrPortAdminStatus          INTEGER,
    vgRptrPortOperStatus           INTEGER,
    vgRptrPortSupportedPromiscMode INTEGER,
    vgRptrPortSupportedCascadeMode INTEGER,
    vgRptrPortAllowedTrainType     INTEGER,
    vgRptrPortLastTrainConfig      OCTET STRING,
    vgRptrPortTrainingResult       OCTET STRING,
    vgRptrPortPriorityEnable       TruthValue,
    vgRptrPortRptrInfoIndex        Integer32
}

vgRptrPortIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object identifies the port within the group
        for which this entry contains information. This
        identifies the port independently from the
        repeater it may be attached to. The numbering
        scheme for ports is implementation specific;
        however, this value can never be greater than
        vgRptrGroupPortCapacity for the associated group."
    REFERENCE
        "IEEE Standard 802.12-1995, 13.2.4.5.1,
        aPortID."
    ::= { vgRptrBasicPortEntry 1 }

vgRptrPortType OBJECT-TYPE
    SYNTAX      INTEGER {
        cascadeExternal(1),
        cascadeInternal(2),
        localExternal(3),
        localInternal(4)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Describes the type of port. One of the
        following:

        cascadeExternal - Port is an uplink with
                        physical connections which
                        are externally visible
        cascadeInternal - Port is an uplink with

```

physical connections which are not externally visible, such as a connection to an internal backplane in a chassis

localExternal - Port is a downlink or local port with externally visible connections

localInternal - Port is a downlink or local port with connections which are not externally visible, such as a connection to an internal agent

'internal' is used to identify ports which place traffic into the repeater, but do not have any external connections. Note that both DTE and cascaded repeater downlinks are considered 'local' ports."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1, aPortType."

::= { vGRptrBasicPortEntry 2 }

vGRptrPortAdminStatus OBJECT-TYPE

SYNTAX INTEGER {
enabled(1),
disabled(2)
}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Port enable/disable function. Enabling a disabled port will cause training to be initiated by the training initiator (the slave mode device) on the link. Setting this object to disabled(2) disables the port.

A disabled port neither transmits nor receives. Once disabled, a port must be explicitly enabled to restore operation. A port which is disabled when power is lost or when a reset is exerted shall remain disabled when normal operation resumes.

The value of this object should be preserved across repeater resets and power failures."

REFERENCE

```

        "IEEE Standard 802.12-1995, 13.2.4.5.1,
        aPortAdministrativeState."
 ::= { vgRptrBasicPortEntry 3 }

```

vgRptrPortOperStatus OBJECT-TYPE

```

SYNTAX      INTEGER {
                active(1),
                inactive(2),
                training(3)
            }

```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Current status for the port as specified by the PORT_META_STATE in the port process module of clause 12 [IEEE Std 802.12].

During initialization or any link warning conditions, vgRptrPortStatus will be 'inactive(2)'.

When Training_Up is received by the repeater on a local port (or when Training_Down is received on a cascade port), vgRptrPortStatus will change to 'training(3)' and vgRptrTrainingResult can be monitored to see the detailed status regarding training.

When 24 consecutive good FCS packets are exchanged and the configuration bits are OK, vgRptrPortStatus will change to 'active(1)'.

A disabled port shall have a port status of 'inactive(2)'."

REFERENCE

```

        "IEEE Standard 802.12, 13.2.4.5.1,
        aPortStatus."
 ::= { vgRptrBasicPortEntry 4 }

```

vgRptrPortSupportedPromiscMode OBJECT-TYPE

```

SYNTAX      INTEGER {
                singleModeOnly(1),
                singleOrPromiscMode(2),
                promiscModeOnly(3)
            }

```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object describes whether the port hardware is capable of supporting promiscuous mode, single address mode (i.e., repeater filters unicasts not addressed to the end station attached to this port), or both. A port for which `vgRptrPortType` is equal to 'cascadeInternal' or 'cascadeExternal' will always have a value of 'promiscModeOnly' for this object."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1, `aSupportedPromiscMode`."

::= { `vgRptrBasicPortEntry` 5 }

`vgRptrPortSupportedCascadeMode` OBJECT-TYPE

SYNTAX INTEGER {
 endNodesOnly(1),
 endNodesOrRepeaters(2),
 cascadePort(3)
 }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object describes whether the port hardware is capable of supporting cascaded repeaters, end nodes, or both. A port for which `vgRptrPortType` is equal to 'cascadeInternal' or 'cascadeExternal' will always have a value of 'cascadePort' for this object."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1, `aSupportedCascadeMode`."

::= { `vgRptrBasicPortEntry` 6 }

`vgRptrPortAllowedTrainType` OBJECT-TYPE

SYNTAX INTEGER {
 allowEndNodesOnly(1),
 allowPromiscuousEndNodes(2),
 allowEndNodesOrRepeaters(3),
 allowAnything(4)
 }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This security object is set by the network manager to configure what type of device is permitted to connect to the port. One of the following values:

<code>allowEndNodesOnly</code>	- only non-promiscuous end nodes permitted.
<code>allowPromiscuousEndNodes</code>	- promiscuous or non-promiscuous end nodes permitted
<code>allowEndNodesOrRepeaters</code>	- repeaters or non-promiscuous end nodes permitted
<code>allowAnything</code>	- repeaters, promiscuous or non-promiscuous end nodes permitted

For a port for which `vgRptrPortType` is equal to `'cascadeInternal'` or `'cascadeExternal'`, the corresponding instance of this object may not be set to `'allowEndNodesOnly'` or `'allowPromiscuousEndNodes'`.

The agent must reject a SET of this object if the value includes no capabilities that are supported by this port's hardware, as defined by the values of the corresponding instances of `vgRptrPortSupportedPromiscMode` and `vgRptrPortSupportedCascadeMode`.

Note that `vgRptrPortSupportPromiscMode` and `vgRptrPortSupportedCascadeMode` represent what the port hardware is capable of supporting. `vgRptrPortAllowedTrainType` is used for setting an administrative policy for a port. The actual set of training configurations that will be allowed to succeed on a port is the intersection of what the hardware will support and what is administratively allowed. The above requirement on what values may be set to this object says that the intersection of what is supported and what is allowed must be non-empty. In other words, it must not result in a situation in which nothing would be allowed to train on that port. However, a value can be set to this object as long as the combination of this object and what is supported by the hardware would still leave at least one configuration that could successfully train on the port.

The value of this object should be preserved across repeater resets and power failures."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1,
aAllowableTrainingType."
::= { vgRptrBasicPortEntry 7 }

vgRptrPortLastTrainConfig OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(2))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object is a 16 bit field. For local ports, this object contains the requested configuration field from the most recent error-free training request frame sent by the device connected to the port. For cascade ports, this object contains the responder's allowed configuration field from the most recent error-free training response frame received in response to training initiated by this repeater. The format of the current version of this field is described in section 3.2. Please refer to the most recent version of the IEEE 802.12 standard for the most up-to-date definition of the format of this object."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1,
aLastTrainingConfig."
::= { vgRptrBasicPortEntry 8 }

vgRptrPortTrainingResult OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(3))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This 18 bit field is used to indicate the result of training. It contains two bits which indicate if error-free training frames have been received, and it also contains the 16 bits of the allowed configuration field from the most recent error-free training response frame on the port.

First Octet: Second and Third Octets:
7 6 5 4 3 2 1 0

```
+---+---+---+---+---+---+---+
|0|0|0|0|0|0|V|G| allowed configuration field |
+---+---+---+---+---+---+---+
```

- V: Valid: set when at least one error-free training frame has been received. Indicates the 16 training configuration bits in `vgRptrPortLastTrainConfig` and `vgRptrPortTrainingResult` contain valid information. This bit is cleared when `vgRptrPortStatus` transitions to the 'inactive' or 'training' state.
- G: LinkGood: indicates the link hardware is OK. Set if 24 consecutive error-free training packets have been exchanged. Cleared when a training packet with errors is received, or when `vgRptrPortStatus` transitions to the 'inactive' or 'training' state.

The format of the current version of the allowed configuration field is described in section 3.2. Please refer to the most recent version of the IEEE 802.12 standard for the most up-to-date definition of the format of this field.

If the port is in training, a management station can examine this object to see if any training packets have been passed successfully. If there have been any good training packets, the Valid bit will be set and the management station can examine the allowed configuration field to see if there is a duplicate address, configuration, or security problem.

Note that on a repeater local port, this repeater generates the training response bits, while on a cascade port, the device at the upper end of the link originated the training response bits."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1, `aTrainingResult`."

::= { `vgRptrBasicPortEntry` 9 }

`vgRptrPortPriorityEnable` OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"A configuration flag used to determine whether the repeater will service high priority requests received on the port as high priority or normal priority. When 'false', high priority requests

on this port will be serviced as normal priority.

The setting of this object has no effect on a cascade port. Also note that the setting of this object has no effect on a port connected to a cascaded repeater. In both of these cases, this setting is treated as always 'true'. The value 'false' only has an effect when the port is a localInternal or localExternal port connected to an end node.

The value of this object should be preserved across repeater resets and power failures."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1, aPriorityEnable."

::= { vgRptrBasicPortEntry 10 }

vgRptrPortRptrInfoIndex OBJECT-TYPE

SYNTAX Integer32 (0..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object identifies the repeater that this port is currently mapped to. The repeater identified by a particular value of this object is the same as that identified by the same value of vgRptrInfoIndex. A value of zero indicates that this port is not currently mapped to any repeater."

::= { vgRptrBasicPortEntry 11 }

vgRptrMonitor OBJECT IDENTIFIER ::= { vgRptrObjects 2 }

vgRptrMonRepeater OBJECT IDENTIFIER ::= { vgRptrMonitor 1 }

vgRptrMonitorTable OBJECT-TYPE

SYNTAX SEQUENCE OF VgRptrMonitorEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A table of performance and error statistics for each repeater in the system. The instance of the vgRptrInfoLastChange associated with a repeater is used to indicate possible discontinuities of the counters in this table that are associated with the same repeater."

```
::= { vGRptrMonRepeater 1 }
```

```
vGRptrMonitorEntry OBJECT-TYPE
```

```
SYNTAX      VGRptrMonitorEntry
```

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

```
STATUS      current
```

```
DESCRIPTION
```

```
    "An entry in the table, containing statistics  
    for a single repeater."
```

```
INDEX       { vGRptrInfoIndex }
```

```
::= { vGRptrMonitorTable 1 }
```

```
VGRptrMonitorEntry ::=
```

```
SEQUENCE {
```

```
    vGRptrMonTotalReadableFrames      Counter32,
```

```
    vGRptrMonTotalReadableOctets      Counter32,
```

```
    vGRptrMonReadableOctetRollovers   Counter32,
```

```
    vGRptrMonHCTotalReadableOctets    Counter64,
```

```
    vGRptrMonTotalErrors               Counter32
```

```
}
```

```
vGRptrMonTotalReadableFrames OBJECT-TYPE
```

```
SYNTAX      Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
    "The total number of good frames of valid frame  
    length that have been received on all ports in  
    this repeater.  If an implementation cannot  
    obtain a count of frames as seen by the repeater  
    itself, this counter may be implemented as the  
    summation of the values of the  
    vGRptrPortReadableFrames counters for all of the  
    ports in this repeater.
```

```
    This counter may experience a discontinuity when  
    the value of the corresponding instance of  
    vGRptrInfoLastChange changes."
```

```
::= { vGRptrMonitorEntry 1 }
```

```
vGRptrMonTotalReadableOctets OBJECT-TYPE
```

```
SYNTAX      Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
    "The total number of octets contained in good  
    frames that have been received on all ports in  
    this repeater.  If an implementation cannot
```

obtain a count of octets as seen by the repeater itself, this counter may be implemented as the summation of the values of the `vgRptrPortReadableOctets` counters for all of the ports in this repeater.

Note that this counter can roll over very quickly. A management station is advised to also poll the `vgRptrReadableOctetRollovers` object, or to use the 64-bit counter defined by `vgRptrMonHCTotalReadableOctets` instead of the two 32-bit counters.

This two-counter mechanism is provided for those network management protocols that do not support 64-bit counters (e.g. SNMPv1). Note that retrieval of these two counters in the same PDU is NOT guaranteed to be atomic.

This counter may experience a discontinuity when the value of the corresponding instance of `vgRptrInfoLastChange` changes."

::= { `vgRptrMonitorEntry 2` }

`vgRptrMonReadableOctetRollovers` OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of times that the associated instance of the `vgRptrMonTotalReadableOctets` counter has rolled over.

This two-counter mechanism is provided for those network management protocols that do not support 64-bit counters (e.g. SNMPv1). Note that retrieval of these two counters in the same PDU is NOT guaranteed to be atomic.

This counter may experience a discontinuity when the value of the corresponding instance of `vgRptrInfoLastChange` changes."

::= { `vgRptrMonitorEntry 3` }

`vgRptrMonHCTotalReadableOctets` OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of octets contained in good frames that have been received on all ports in this repeater. If an implementation cannot obtain a count of octets as seen by the repeater itself, this counter may be implemented as the summation of the values of the vgRptrPortHCReadableOctets counters for all of the ports in this repeater.

This counter is a 64 bit version of vgRptrMonTotalReadableOctets. It should be used by Network Management protocols which support 64 bit counters (e.g. SNMPv2).

This counter may experience a discontinuity when the value of the corresponding instance of vgRptrInfoLastChange changes."

::= { vgRptrMonitorEntry 4 }

vgRptrMonTotalErrors OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of errors which have occurred on all of the ports in this repeater. If an implementation cannot obtain a count of these errors as seen by the repeater itself, this counter may be implemented as the summation of the values of the vgRptrPortIPMFrames, vgRptrPortOversizeFrames, and vgRptrPortDataErrorFrames counters for all of the ports in this repeater.

This counter may experience a discontinuity when the value of the corresponding instance of vgRptrInfoLastChange changes."

::= { vgRptrMonitorEntry 5 }

vgRptrMonGroup OBJECT IDENTIFIER ::= { vgRptrMonitor 2 }
-- Currently unused

vgRptrMonPort OBJECT IDENTIFIER ::= { vgRptrMonitor 3 }

vgRptrMonPortTable OBJECT-TYPE

SYNTAX SEQUENCE OF VgRptrMonPortEntry

MAX-ACCESS not-accessible

```

STATUS      current
DESCRIPTION
    "A table of performance and error statistics for
    the ports.  The columnar object
    vgRptrPortLastChange is used to indicate possible
    discontinuities of counter type columnar objects
    in this table."
 ::= { vgRptrMonPort 1 }

vgRptrMonPortEntry OBJECT-TYPE
SYNTAX      VgRptrMonPortEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "An entry in the vgRptrMonPortTable, containing
    performance and error statistics for a single
    port."
INDEX       { vgRptrGroupIndex, vgRptrPortIndex }
 ::= { vgRptrMonPortTable 1 }

VgRptrMonPortEntry ::=
SEQUENCE {
    vgRptrPortReadableFrames      Counter32,
    vgRptrPortReadableOctets      Counter32,
    vgRptrPortReadOctetRollovers  Counter32,
    vgRptrPortHCReadableOctets    Counter64,
    vgRptrPortUnreadableOctets    Counter32,
    vgRptrPortUnreadOctetRollovers Counter32,
    vgRptrPortHCUnreadableOctets  Counter64,
    vgRptrPortHighPriorityFrames   Counter32,
    vgRptrPortHighPriorityOctets   Counter32,
    vgRptrPortHighPriOctetRollovers Counter32,
    vgRptrPortHCHighPriorityOctets Counter64,
    vgRptrPortNormPriorityFrames   Counter32,
    vgRptrPortNormPriorityOctets   Counter32,
    vgRptrPortNormPriOctetRollovers Counter32,
    vgRptrPortHCNormPriorityOctets Counter64,
    vgRptrPortBroadcastFrames     Counter32,
    vgRptrPortMulticastFrames     Counter32,
    vgRptrPortNullAddressedFrames Counter32,
    vgRptrPortIPMFrames           Counter32,
    vgRptrPortOversizeFrames       Counter32,
    vgRptrPortDataErrorFrames      Counter32,
    vgRptrPortPriorityPromotions    Counter32,
    vgRptrPortTransitionToTrainings Counter32,
    vgRptrPortLastChange           TimeStamp
}

```


vgRptrPortReadableFrames OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"This object is the number of good frames of valid frame length that have been received on this port. This counter is incremented by one for each frame received on the port which is not counted by any of the following error counters: vgRptrPortIPMFrames, vgRptrPortOversizeFrames, vgRptrPortNullAddressedFrames, or vgRptrPortDataErrorFrames.

This counter may experience a discontinuity when the value of the corresponding instance of vgRptrPortLastChange changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1, aReadableFramesReceived."

::= { vgRptrMonPortEntry 1 }

vgRptrPortReadableOctets OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"This object is a count of the number of octets contained in good frames that have been received on this port. This counter is incremented by OctetCount for each frame received on this port which has been determined to be a readable frame (i.e. each frame counted by vgRptrPortReadableFrames).

Note that this counter can roll over very quickly. A management station is advised to also poll the vgRptrPortReadOctetRollovers object, or to use the 64-bit counter defined by vgRptrPortHCReadableOctets instead of the two 32-bit counters.

This two-counter mechanism is provided for those network management protocols that do not support 64-bit counters (e.g. SNMPv1). Note that retrieval of these two counters in the same PDU is NOT guaranteed to be atomic.

This counter may experience a discontinuity when the value of the corresponding instance of `vgRptrPortLastChange` changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1, `aReadableOctetsReceived`."

::= { `vgRptrMonPortEntry 2` }

`vgRptrPortReadOctetRollovers` OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object is a count of the number of times that the associated instance of the `vgRptrPortReadableOctets` counter has rolled over.

This two-counter mechanism is provided for those network management protocols that do not support 64-bit counters (e.g. SNMPv1). Note that retrieval of these two counters in the same PDU is NOT guaranteed to be atomic.

This counter may experience a discontinuity when the value of the corresponding instance of `vgRptrPortLastChange` changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1, `aReadableOctetsReceived`."

::= { `vgRptrMonPortEntry 3` }

`vgRptrPortHCReadableOctets` OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object is a count of the number of octets contained in good frames that have been received on this port. This counter is incremented by `OctetCount` for each frame received on this port which has been determined to be a readable frame (i.e. each frame counted by `vgRptrPortReadableFrames`).

This counter is a 64 bit version of `vgRptrPortReadableOctets`. It should be used by Network Management protocols which support 64 bit counters (e.g. SNMPv2).

This counter may experience a discontinuity when the value of the corresponding instance of `vgRptrPortLastChange` changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1,
 `aReadableOctetsReceived`."
 ::= { `vgRptrMonPortEntry` 4 }

`vgRptrPortUnreadableOctets` OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"This object is a count of the number of octets contained in invalid frames that have been received on this port. This counter is incremented by `OctetCount` for each frame received on this port which is counted by `vgRptrPortIPMFrames`, `vgRptrPortOversizeFrames`, `vgRptrPortNullAddressedFrames`, or `vgRptrPortDataErrorFrames`. This counter can be combined with `vgRptrPortReadableOctets` to calculate network utilization.

Note that this counter can roll over very quickly. A management station is advised to also poll the `vgRptrPortUnreadOctetRollovers` object, or to use the 64-bit counter defined by `vgRptrPortHCUnreadableOctets` instead of the two 32-bit counters.

This two-counter mechanism is provided for those network management protocols that do not support 64-bit counters (e.g. SNMPv1). Note that retrieval of these two counters in the same PDU is NOT guaranteed to be atomic.

This counter may experience a discontinuity when the value of the corresponding instance of `vgRptrPortLastChange` changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1,
 `aOctetsInUnreadableFramesRcvd`."
 ::= { `vgRptrMonPortEntry` 5 }

`vgRptrPortUnreadOctetRollovers` OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object is a count of the number of times that the associated instance of the vgRptrPortUnreadableOctets counter has rolled over.

This two-counter mechanism is provided for those network management protocols that do not support 64-bit counters (e.g. SNMPv1). Note that retrieval of these two counters in the same PDU is NOT guaranteed to be atomic.

This counter may experience a discontinuity when the value of the corresponding instance of vgRptrPortLastChange changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1, aOctetsInUnreadableFramesRcvd."

::= { vgRptrMonPortEntry 6 }

vgRptrPortHCUnreadableOctets OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object is a count of the number of octets contained in invalid frames that have been received on this port. This counter is incremented by OctetCount for each frame received on this port which is counted by vgRptrPortIPMFrames, vgRptrPortOversizeFrames, vgRptrPortNullAddressedFrames, or vgRptrPortDataErrorFrames. This counter can be combined with vgRptrPortHCReadableOctets to calculate network utilization.

This counter is a 64 bit version of vgRptrPortUnreadableOctets. It should be used by Network Management protocols which support 64 bit counters (e.g. SNMPv2).

This counter may experience a discontinuity when the value of the corresponding instance of vgRptrPortLastChange changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1, aOctetsInUnreadableFramesRcvd."

::= { vgRptrMonPortEntry 7 }

vgRptrPortHighPriorityFrames OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object is a count of high priority frames that have been received on this port. This counter is incremented by one for each high priority frame received on this port. This counter includes both good and bad high priority frames, as well as high priority training frames. This counter does not include normal priority frames which were priority promoted.

This counter may experience a discontinuity when the value of the corresponding instance of vgRptrPortLastChange changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1, aHighPriorityFramesReceived."

::= { vgRptrMonPortEntry 8 }

vgRptrPortHighPriorityOctets OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object is a count of the number of octets contained in high priority frames that have been received on this port. This counter is incremented by OctetCount for each frame received on this port which is counted by vgRptrPortHighPriorityFrames.

Note that this counter can roll over very quickly. A management station is advised to also poll the vgRptrPortHighPriOctetRollovers object, or to use the 64-bit counter defined by vgRptrPortHCHighPriorityOctets instead of the two 32-bit counters.

This two-counter mechanism is provided for those network management protocols that do not support 64-bit counters (e.g. SNMPv1). Note that retrieval of these two counters in the same PDU is NOT guaranteed to be atomic.

This counter may experience a discontinuity when the value of the corresponding instance of `vgRptrPortLastChange` changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1, `aHighPriorityOctetsReceived`."

::= { `vgRptrMonPortEntry` 9 }

`vgRptrPortHighPriOctetRollovers` OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object is a count of the number of times that the associated instance of the `vgRptrPortHighPriorityOctets` counter has rolled over.

This two-counter mechanism is provided for those network management protocols that do not support 64-bit counters (e.g. SNMPv1). Note that retrieval of these two counters in the same PDU is NOT guaranteed to be atomic.

This counter may experience a discontinuity when the value of the corresponding instance of `vgRptrPortLastChange` changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1, `aHighPriorityOctetsReceived`."

::= { `vgRptrMonPortEntry` 10 }

`vgRptrPortHCHighPriorityOctets` OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object is a count of the number of octets contained in high priority frames that have been received on this port. This counter is incremented by `OctetCount` for each frame received on this port which is counted by `vgRptrPortHighPriorityFrames`.

This counter is a 64 bit version of `vgRptrPortHighPriorityOctets`. It should be used by Network Management protocols which support 64 bit counters (e.g. SNMPv2).

This counter may experience a discontinuity when the value of the corresponding instance of `vgRptrPortLastChange` changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1, `aHighPriorityOctetsReceived`."

::= { `vgRptrMonPortEntry` 11 }

`vgRptrPortNormPriorityFrames` OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object is a count of normal priority frames that have been received on this port. This counter is incremented by one for each normal priority frame received on this port. This counter includes both good and bad normal priority frames, as well as normal priority training frames and normal priority frames which were priority promoted.

This counter may experience a discontinuity when the value of the corresponding instance of `vgRptrPortLastChange` changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1, `aNormalPriorityFramesReceived`."

::= { `vgRptrMonPortEntry` 12 }

`vgRptrPortNormPriorityOctets` OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object is a count of the number of octets contained in normal priority frames that have been received on this port. This counter is incremented by `OctetCount` for each frame received on this port which is counted by `vgRptrPortNormPriorityFrames`.

Note that this counter can roll over very quickly. A management station is advised to also poll the `vgRptrPortNormPriOctetRollovers` object, or to use the 64-bit counter defined by `vgRptrPortHCNormPriorityOctets` instead of the two 32-bit counters.

This two-counter mechanism is provided for those network management protocols that do not support 64-bit counters (e.g. SNMPv1). Note that retrieval of these two counters in the same PDU is NOT guaranteed to be atomic.

This counter may experience a discontinuity when the value of the corresponding instance of `vgRptrPortLastChange` changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1, `aNormalPriorityOctetsReceived`.
`::= { vgRptrMonPortEntry 13 }`

`vgRptrPortNormPriOctetRollovers` OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object is a count of the number of times that the associated instance of the `vgRptrPortNormPriorityOctets` counter has rolled over.

This two-counter mechanism is provided for those network management protocols that do not support 64-bit counters (e.g. SNMPv1). Note that retrieval of these two counters in the same PDU is NOT guaranteed to be atomic.

This counter may experience a discontinuity when the value of the corresponding instance of `vgRptrPortLastChange` changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1, `aNormalPriorityOctetsReceived`.
`::= { vgRptrMonPortEntry 14 }`

`vgRptrPortHCNormPriorityOctets` OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object is a count of the number of octets contained in normal priority frames that have been received on this port. This counter is incremented by `OctetCount` for each frame received

on this port which is counted by
vgRptrPortNormPriorityFrames.

This counter is a 64 bit version of
vgRptrPortNormPriorityOctets. It should be used
by Network Management protocols which support
64 bit counters (e.g. SNMPv2).

This counter may experience a discontinuity when
the value of the corresponding instance of
vgRptrPortLastChange changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1,
aNormalPriorityOctetsReceived."

::= { vgRptrMonPortEntry 15 }

vgRptrPortBroadcastFrames OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object is a count of broadcast packets that
have been received on this port. This counter is
incremented by one for each readable frame
received on this port whose destination MAC
address is the broadcast address. Frames
counted by this counter are also counted by
vgRptrPortReadableFrames.

This counter may experience a discontinuity when
the value of the corresponding instance of
vgRptrPortLastChange changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1,
aBroadcastFramesReceived."

::= { vgRptrMonPortEntry 16 }

vgRptrPortMulticastFrames OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object is a count of multicast packets that
have been received on this port. This counter is
incremented by one for each readable frame
received on this port whose destination MAC
address has the group address bit set, but is not
the broadcast address. Frames counted by this

counter are also counted by
 vgRptrPortReadableFrames, but not by
 vgRptrPortBroadcastFrames. Note that when the
 value of the instance vgRptrInfoCurrentFramingType
 for the repeater that this port is associated
 with is equal to 'frameType88025', this count
 includes packets addressed to functional
 addresses.

This counter may experience a discontinuity when
 the value of the corresponding instance of
 vgRptrPortLastChange changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1,
 aMulticastFramesReceived."

::= { vgRptrMonPortEntry 17 }

vgRptrPortNullAddressedFrames OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object is a count of null addressed packets
 that have been received on this port. This
 counter is incremented by one for each frame
 received on this port with a destination MAC
 address consisting of all zero bits. Both void
 and training frames are included in this
 counter.

This counter may experience a discontinuity when
 the value of the corresponding instance of
 vgRptrPortLastChange changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1,
 aNullAddressedFramesReceived."

::= { vgRptrMonPortEntry 18 }

vgRptrPortIPMFrames OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object is a count of the number of frames
 that have been received on this port with an
 invalid packet marker and no PMI errors. A
 repeater will write an invalid packet marker to
 the end of a frame containing errors as it is

forwarded through the repeater to the other ports. This counter is incremented by one for each frame received on this port which has had an invalid packet marker added to the end of the frame.

This counter indicates problems occurring in the domain of other repeaters, as opposed to problems with cables or devices directly attached to this repeater.

This counter may experience a discontinuity when the value of the corresponding instance of `vgRptrPortLastChange` changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1, `aIPMFramesReceived`."

::= { `vgRptrMonPortEntry` 19 }

`vgRptrPortOversizeFrames` OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object is a count of oversize frames received on this port. This counter is incremented by one for each frame received on this port whose `OctetCount` is larger than the maximum legal frame size.

The frame size which causes this counter to increment is dependent on the current value of `vgRptrInfoCurrentFramingType` for the repeater that the port is associated with. When `vgRptrInfoCurrentFramingType` is equal to `frameType88023` this counter will increment for frames that are 1519 octets or larger. When `vgRptrInfoCurrentFramingType` is equal to `frameType88025` this counter will increment for frames that are 4521 octets or larger.

This counter may experience a discontinuity when the value of the corresponding instance of `vgRptrPortLastChange` changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1, `aOversizeFramesReceived`."

::= { `vgRptrMonPortEntry` 20 }

vgRptrPortDataErrorFrames OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"This object is a count of errored frames received on this port. This counter is incremented by one for each frame received on this port with any of the following errors: bad FCS (with no IPM), PMI errors (excluding frames with an IPM error as the only PMI error), or undersize (with no IPM). Does not include packets counted by vgRptrPortIPMFrames, vgRptrPortOversizeFrames, or vgRptrPortNullAddressedFrames.

This counter indicates problems with cables or devices directly connected to this repeater, while vgRptrPortIPMFrames indicates problems occurring in the domain of other repeaters.

This counter may experience a discontinuity when the value of the corresponding instance of vgRptrPortLastChange changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1,
aDataErrorFramesReceived."
::= { vgRptrMonPortEntry 21 }

vgRptrPortPriorityPromotions OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"This counter is incremented by one each time the priority promotion timer has expired on this port and a normal priority frame is priority promoted.

This counter may experience a discontinuity when the value of the corresponding instance of vgRptrPortLastChange changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1,
aPriorityPromotions."
::= { vgRptrMonPortEntry 22 }

vgRptrPortTransitionToTrainings OBJECT-TYPE

```

SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "This counter is incremented by one each time the
    vgRptrPortStatus object for this port transitions
    into the 'training' state.

    This counter may experience a discontinuity when
    the value of the corresponding instance of
    vgRptrPortLastChange changes."
REFERENCE
    "IEEE Standard 802.12-1995, 13.2.4.5.1,
    aTransitionsIntoTraining."
::= { vgRptrMonPortEntry 23 }

vgRptrPortLastChange OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The value of sysUpTime when the last of the
        following occurred:
        1) the agent cold- or warm-started;
        2) the row for the port was created
           (such as when a device or module was
           added to the system); or
        3) any condition that would cause one of
           the counters for the row to experience
           a discontinuity."
    ::= { vgRptrMonPortEntry 24 }

vgRptrAddrTrack    OBJECT IDENTIFIER ::= { vgRptrObjects 3 }

vgRptrAddrTrackRptr
    OBJECT IDENTIFIER ::= { vgRptrAddrTrack 1 }

-- Currently unused

vgRptrAddrTrackGroup
    OBJECT IDENTIFIER ::= { vgRptrAddrTrack 2 }
-- Currently unused

vgRptrAddrTrackPort
    OBJECT IDENTIFIER ::= { vgRptrAddrTrack 3 }

vgRptrAddrTrackTable OBJECT-TYPE

```

```

SYNTAX      SEQUENCE OF VgRptrAddrTrackEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "Table of address mapping information about the
    ports."
 ::= { vgRptrAddrTrackPort 1 }

```

```

vgRptrAddrTrackEntry OBJECT-TYPE
SYNTAX      VgRptrAddrTrackEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "An entry in the table, containing address mapping
    information about a single port."
INDEX       { vgRptrGroupIndex, vgRptrPortIndex }
 ::= { vgRptrAddrTrackTable 1 }

```

```

VgRptrAddrTrackEntry ::=
SEQUENCE {
    vgRptrAddrLastTrainedAddress    OCTET STRING,
    vgRptrAddrTrainedAddrChanges    Counter32,
    vgRptrRptrDetectedDupAddress    TruthValue,
    vgRptrMgrDetectedDupAddress     TruthValue
}

```

```

vgRptrAddrLastTrainedAddress OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE(0 | 6))
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "This object is the MAC address of the last
    station which succeeded in training on this port.
    A cascaded repeater may train using the null
    address.  If no stations have succeeded in
    training on this port since the agent began
    monitoring the port activity, the agent shall
    return a string of length zero."
REFERENCE
    "IEEE Standard 802.12-1995, 13.2.4.5.1,
    aLastTrainedAddress."
 ::= { vgRptrAddrTrackEntry 1 }

```

```

vgRptrAddrTrainedAddrChanges OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current

```

DESCRIPTION

"This counter is incremented by one for each time that the vgRptrAddrLastTrainedAddress object for this port changes."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1, aTrainedAddressChanges."

::= { vgRptrAddrTrackEntry 2 }

vgRptrRptrDetectedDupAddress OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object is used to indicate that the repeater detected an error-free training frame on this port with a non-null source MAC address which matches the value of vgRptrAddrLastTrainedAddress of another active port in the same repeater. This is reset to 'false' when an error-free training frame is received with a non-null source MAC address which does not match vgRptrAddrLastTrainedAddress of another port which is active in the same repeater."

For the cascade port, this object will be 'true' if the 'D' bit in the most recently received error-free training response frame was set, indicating the device at the other end of the link believes that this repeater's cascade port is using a duplicate address. This may be because the device at the other end of the link detected a duplicate address itself, or, if the other device is also a repeater, it could be because vgRptrMgrDetectedDupAddress was set to 'true' on the port that this repeater's cascade port is connected to."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1, aLocalRptrDetectedDupAddr."

::= { vgRptrAddrTrackEntry 3 }

vgRptrMgrDetectedDupAddress OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object can be set by a management station

when it detects that there is a duplicate MAC address. This object is OR'd with `vgRptrRptrDetectedDupAddress` to form the value of the 'D' bit in training response frames on this port.

The purpose of this object is to provide a means for network management software to inform an end station that it is using a duplicate station address. Setting this object does not affect the current state of the link; the end station will not be informed of the duplicate address until it retrains for some reason. Note that regardless of its station address, the end station will not be able to train successfully until the network management software has set this object back to 'false'. Although this object exists on cascade ports, it does not perform any function since this repeater is the initiator of training on a cascade port."

REFERENCE

"IEEE Standard 802.12-1995, 13.2.4.5.1,
`aCentralMgmtDetectedDupAddr`."
`::= { vgRptrAddrTrackEntry 4 }`

`vgRptrTraps` OBJECT IDENTIFIER ::= { `vgRptrMIB 2` }
`vgRptrTrapPrefix` OBJECT IDENTIFIER ::= { `vgRptrTraps 0` }

`vgRptrHealth` NOTIFICATION-TYPE

OBJECTS { `vgRptrInfoOperStatus` }
 STATUS current

DESCRIPTION

"A `vgRptrHealth` trap conveys information related to the operational state of a repeater. This trap is sent when the value of an instance of `vgRptrInfoOperStatus` changes. The `vgRptrHealth` trap is not sent as a result of powering up a repeater.

The `vgRptrHealth` trap must contain the instance of the `vgRptrInfoOperStatus` object associated with the affected repeater.

The agent must throttle the generation of consecutive `vgRptrHealth` traps so that there is at least a five-second gap between traps of this type. When traps are throttled, they are dropped,

not queued for sending at a future time. (Note that 'generating' a trap means sending to all configured recipients.)"

REFERENCE

"IEEE 802.12, Layer Management, 13.2.4.2.3, nRepeaterHealth."

::= { vgRptrTrapPrefix 1 }

vgRptrResetEvent NOTIFICATION-TYPE

OBJECTS { vgRptrInfoOperStatus }

STATUS current

DESCRIPTION

"A vgRptrResetEvent trap conveys information related to the operational state of a repeater. This trap is sent on completion of a repeater reset action. A repeater reset action is defined as a transition to its initial state as specified in clause 12 [IEEE Std 802.12] when triggered by a management command.

The vgRptrResetEvent trap is not sent when the agent restarts and sends an SNMP coldStart or warmStart trap.

The vgRptrResetEvent trap must contain the instance of the vgRptrInfoOperStatus object associated with the affected repeater.

The agent must throttle the generation of consecutive vgRptrResetEvent traps so that there is at least a five-second gap between traps of this type. When traps are throttled, they are dropped, not queued for sending at a future time. (Note that 'generating' a trap means sending to all configured recipients.)"

REFERENCE

"IEEE 802.12, Layer Management, 13.2.4.2.3, nRepeaterReset."

::= { vgRptrTrapPrefix 2 }

-- conformance information

vgRptrConformance OBJECT IDENTIFIER ::= { vgRptrMIB 3 }

vgRptrCompliances

OBJECT IDENTIFIER ::= { vgRptrConformance 1 }

vgRptrGroups OBJECT IDENTIFIER ::= { vgRptrConformance 2 }

-- compliance statements

vgRptrCompliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION

"The compliance statement for managed 802.12 repeaters."

MODULE -- this module

MANDATORY-GROUPS { vgRptrConfigGroup,
vgRptrStatsGroup,
vgRptrAddrGroup,
vgRptrNotificationsGroup }

GROUP vgRptrStats64Group

DESCRIPTION

"Implementation of this group is recommended for systems which can support Counter64."

OBJECT vgRptrInfoDesiredFramingType

MIN-ACCESS read-only

DESCRIPTION

"Write access to this object is not required in a repeater system that does not support configuration of framing types."

MODULE SNMP-REPEATER-MIB

GROUP snmpRptrGrpRptrAddrSearch

DESCRIPTION

"Implementation of this group is recommended for systems which have the necessary instrumentation to search all incoming data streams for a particular source MAC address."

::= { vgRptrCompliances 1 }

-- units of conformance

vgRptrConfigGroup OBJECT-GROUP

OBJECTS {

vgRptrInfoMACAddress,
vgRptrInfoCurrentFramingType,
vgRptrInfoDesiredFramingType,
vgRptrInfoFramingCapability,
vgRptrInfoTrainingVersion,
vgRptrInfoOperStatus,
vgRptrInfoReset,
vgRptrInfoLastChange,
vgRptrGroupObjectID,

```

        vgRptrGroupOperStatus,
        vgRptrGroupPortCapacity,
        vgRptrGroupCablesBundled,
        vgRptrPortType,
        vgRptrPortAdminStatus,
        vgRptrPortOperStatus,
        vgRptrPortSupportedPromiscMode,
        vgRptrPortSupportedCascadeMode,
        vgRptrPortAllowedTrainType,
        vgRptrPortLastTrainConfig,
        vgRptrPortTrainingResult,
        vgRptrPortPriorityEnable,
        vgRptrPortRptrInfoIndex
    }
STATUS      current
DESCRIPTION
    "A collection of objects for managing the status
    and configuration of IEEE 802.12 repeaters."
 ::= { vgRptrGroups 1 }

vgRptrStatsGroup OBJECT-GROUP
OBJECTS      {
    vgRptrMonTotalReadableFrames,
    vgRptrMonTotalReadableOctets,
    vgRptrMonReadableOctetRollovers,
    vgRptrMonTotalErrors,
    vgRptrPortReadableFrames,
    vgRptrPortReadableOctets,
    vgRptrPortReadOctetRollovers,
    vgRptrPortUnreadableOctets,
    vgRptrPortUnreadOctetRollovers,
    vgRptrPortHighPriorityFrames,
    vgRptrPortHighPriorityOctets,
    vgRptrPortHighPriOctetRollovers,
    vgRptrPortNormPriorityFrames,
    vgRptrPortNormPriorityOctets,
    vgRptrPortNormPriOctetRollovers,
    vgRptrPortBroadcastFrames,
    vgRptrPortMulticastFrames,
    vgRptrPortNullAddressedFrames,
    vgRptrPortIPMFrames,
    vgRptrPortOversizeFrames,
    vgRptrPortDataErrorFrames,
    vgRptrPortPriorityPromotions,
    vgRptrPortTransitionToTrainings,
    vgRptrPortLastChange
}
STATUS      current

```

```

DESCRIPTION
    "A collection of objects for providing statistics
    for IEEE 802.12 repeaters. Systems which support
    Counter64 should also implement
    vgRptrStats64Group."
 ::= { vgRptrGroups 2 }

vgRptrStats64Group OBJECT-GROUP
    OBJECTS {
        vgRptrMonHCTotalReadableOctets,
        vgRptrPortHCReadableOctets,
        vgRptrPortHCUnreadableOctets,
        vgRptrPortHCHighPriorityOctets,
        vgRptrPortHCNormPriorityOctets
    }
    STATUS current
    DESCRIPTION
        "A collection of objects for providing statistics
        for IEEE 802.12 repeaters in a system that
        supports Counter64."
 ::= { vgRptrGroups 3 }

vgRptrAddrGroup OBJECT-GROUP
    OBJECTS {
        vgRptrAddrLastTrainedAddress,
        vgRptrAddrTrainedAddrChanges,
        vgRptrRptrDetectedDupAddress,
        vgRptrMgrDetectedDupAddress
    }
    STATUS current
    DESCRIPTION
        "A collection of objects for tracking addresses
        on IEEE 802.12 repeaters."
 ::= { vgRptrGroups 4 }

vgRptrNotificationsGroup NOTIFICATION-GROUP
    NOTIFICATIONS {
        vgRptrHealth,
        vgRptrResetEvent
    }
    STATUS current
    DESCRIPTION
        "A collection of notifications used to indicate
        802.12 repeater general status changes."
 ::= { vgRptrGroups 5 }

END

```

4. Acknowledgements

This document was produced by the IETF 100VG-AnyLAN Working Group, whose efforts were greatly advanced by the contributions of the following people:

Paul Chefurka
Bob Faulk
Jeff Johnson
Karen Kimball
David Lapp
Jason Spofford
Kaj Tesink

This document is based on the work of IEEE 802.12.

5. References

- [1] Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), International Organization for Standardization. International Standard 8824 (December, 1987).
- [2] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1902, January 1996.
- [3] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1903, January 1996.
- [4] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1904, January 1996.
- [5] McCloghrie, K. and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets - MIB-II", STD 17, RFC 1213, March 1991.
- [6] IEEE, "Demand Priority Access Method, Physical Layer and Repeater Specifications for 100 Mb/s Operation", IEEE Standard 802.12-1995"

- [7] de Graaf, K., D. Romascanu, D. McMaster, and K. McCloghrie, "Definitions of Managed Objects for IEEE 802.3 Repeater Devices", RFC 2108, 3Com Corporation, Madge Networks (Israel) Ltd., Cisco Systems, Inc., February, 1997.
- [8] McAnally, G., Gilbert, D. and J. Flick, "Conditional Grant of Rights to Specific Hewlett-Packard Patents In Conjunction With the Internet Engineering Task Force's Internet-Standard Network Management Framework", RFC 1988, August 1996.
- [9] Hewlett-Packard Company, US Patents 5,293,635 and 5,421,024.

6. Security Considerations

Certain management information defined in this MIB may be considered sensitive in some network environments. Therefore, authentication of received SNMP requests and controlled access to management information should be employed in such environments. The method for this authentication is a function of the SNMP Administrative Framework, and has not been expanded by this MIB.

Several objects in the `vgRptrConfigGroup` allow write access. Setting these objects can have a serious effect on the operation of the network, including modifying the framing type of the network, resetting the repeater, enabling and disabling individual ports, and modifying the allowed capabilities of end stations attached to each port. It is recommended that implementers seriously consider whether set operations should be allowed without providing, at a minimum, authentication of request origin.

One particular object in this MIB, `vgRptrPortAllowedTrainType`, is considered significant for providing operational security in an 802.12 network. It is recommended that network administrators configure this object to the 'allowEndNodesOnly' value on all ports except ports which the administrator knows are attached to cascaded repeaters or devices which require promiscuous receive capability (bridges, switches, RMON probes, etc.). This will prevent unauthorized users from extending the network (by attaching cascaded repeaters or bridges) without the administrator's knowledge, and will prevent unauthorized end nodes from listening promiscuously to network traffic.

7. Author's Address

John Flick
Hewlett Packard Company
8000 Foothills Blvd. M/S 5556
Roseville, CA 95747-5556

Phone: +1 916 785 4018
Email: johnf@hprnd.rose.hp.com

8. Full Copyright Statement

Copyright (C) The Internet Society (1998). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.

