

OPSAWG  
Internet-Draft  
Intended status: Standards Track  
Expires: April 26, 2013

H. Asai  
Y. Sekiya  
The University of Tokyo  
K. Shima  
IIJ Innovation Institute Inc.  
H. Esaki  
The University of Tokyo  
October 23, 2012

**Management Information Base for the Virtual Machine Monitoring  
draft-asai-vmm-mib-01**

**Abstract**

This document defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, this specifies managed objects that are used for hypervisors (a.k.a. virtual machine managers), virtual resources provided by them, and virtual machines running on them.

**Status of this Memo**

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 26, 2013.

**Copyright Notice**

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">1.1.</a>	Requirements Language . . . . .	<a href="#">3</a>
<a href="#">2.</a>	The SNMP Network Management Framework . . . . .	<a href="#">4</a>
<a href="#">2.1.</a>	Managed Objects on Virtualization Environment . . . . .	<a href="#">5</a>
<a href="#">3.</a>	Definitions . . . . .	<a href="#">7</a>
<a href="#">4.</a>	IANA Considerations . . . . .	<a href="#">27</a>
<a href="#">5.</a>	Security Considerations . . . . .	<a href="#">28</a>
<a href="#">6.</a>	Normative References . . . . .	<a href="#">30</a>
	Authors' Addresses . . . . .	<a href="#">32</a>



## **1. Introduction**

This document defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, this specifies managed objects that are used for hypervisors (a.k.a. virtual machine managers), virtual resources provided by them, and virtual machines running on them. A hypervisor manages multiple virtual machines on a single physical machine by allocating resources to each virtual machine using virtualization technologies. Thus, the MIB objects include information on virtual machines and virtual resources managed by a hypervisor to virtual machines as well as hypervisor's hardware and software information.

### **1.1. Requirements Language**

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 [RFC 2119](#).



## **2. The SNMP Network Management Framework**

The SNMP Network Management Framework presently consists of three major components;

- o An overall architecture, described in [RFC 3411](#) [[RFC3411](#)]
- o Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIV1 and described in STD 16, [RFC 1155](#) [[RFC1155](#)], STD 16, [RFC 1212](#) [[RFC1212](#)] and [RFC 1215](#) [[RFC1215](#)]. The second version, called SMIV2, is described in STD 58, [RFC 2578](#) [[RFC2578](#)], [RFC 2579](#) [[RFC2579](#)] and [RFC 2580](#) [[RFC2580](#)].
- o Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, [RFC 1157](#) [[RFC1157](#)]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in [RFC 1901](#) [[RFC1901](#)] and [RFC 3417](#) [[RFC3417](#)]. The third version of the message protocol is called SNMPv3 and described in [RFC 3412](#) [[RFC3412](#)], [RFC 3414](#) [[RFC3414](#)] and [RFC 3417](#) [[RFC3417](#)].
- o Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, [RFC 1157](#) [[RFC1157](#)]. A second set of protocol operations and associated PDU formats is described in [RFC 3416](#) [[RFC3416](#)].
- o A set of fundamental applications described in [RFC 2573](#) [[RFC2573](#)] and the view-based access control mechanism described in [RFC 2575](#) [[RFC2575](#)].

A more detailed introduction to the current SNMP Management Framework can be found in [RFC 3410](#) [[RFC3410](#)].

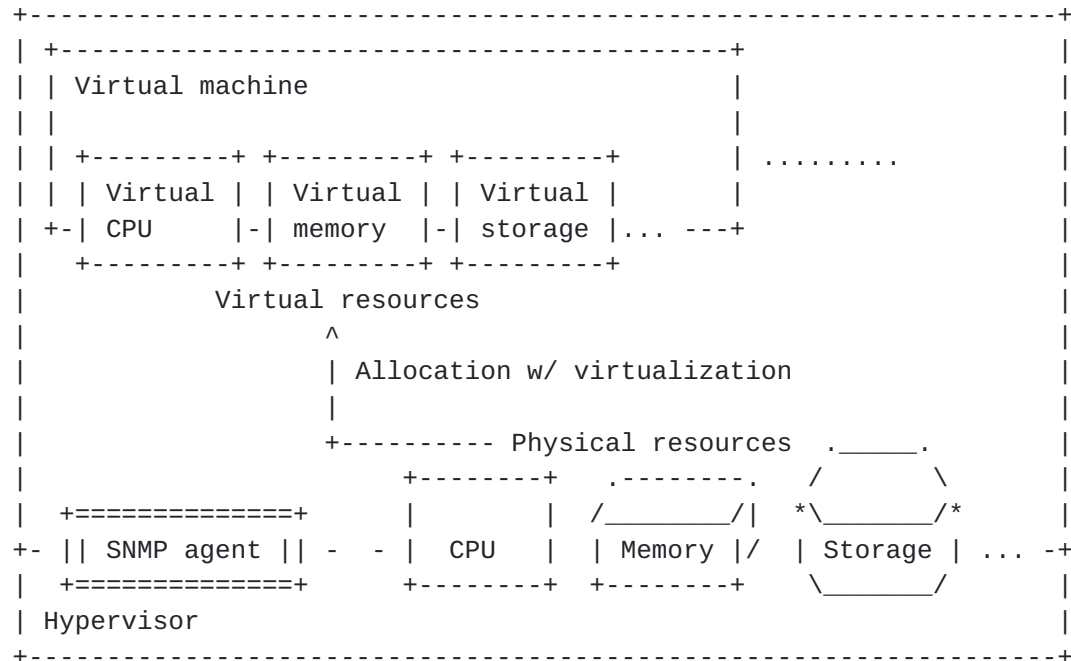
Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This document specifies a MIB module that is compliant to the SMIV2. A MIB conforming to the SMIV1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine readable information in SMIV2 will be converted into textual descriptions in SMIV1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the



MIB.

## 2.1. Managed Objects on Virtualization Environment



A hypervisor allocates resources as virtual devices such as virtual CPU, virtual memory, virtual storage, and virtual network interface to multiple virtual machines from physical resources.

Figure 1: An example of a virtualization environment

A hypervisor allocates resources as virtual devices such as virtual CPUs, virtual memory, virtual storage, and virtual network interface to multiple virtual machines from physical resources.

This document defines objects related to system and software information of a hypervisor, the list of virtual machines running on the hypervisor, and virtual resources allocated by the hypervisor to virtual machines. The virtual resource objects are defined as virtual devices, and this document defines four specific types of virtual devices; processors, memory modules, network interfaces, and storage devices. Note that physical resources are managed in HOST-RESOURCES-MIB [RFC 2790](#) [[RFC2790](#)]. In case that each virtual resource device object has a corresponding parent physical device managed in HOST-RESOURCES-MIB, the object of the virtual resource device contains a pointer to the physical device. The objects related to virtual network interfaces are mapped to the objects managed in IF-MIB [RFC 2863](#) [[RFC2863](#)].





The objects defined this document are managed at a hypervisor and an SNMP agent is launched at the hypervisor to provide access to the objects. The objects are managed from the viewpoint of the operators of hypervisors, but not the operators of virtual machines. For example, the objects do not take into account the actual resource utilization at each virtual machine but the resource allocation from the physical resources. Therefore, `vrNetworkIndex` indicates the virtual interface associated with an interface of a virtual machine at the hypervisor, and consequently, the `in` and `out` directions denote `from the virtual machine to the hypervisor` and `from the hypervisor to the virtual machine`, respectively. Moreover, `vrStorageAllocatedSize` denotes the size allocated in the hypervisor, but not the size actually used by the operating system on the virtual machine. This means that `vrStorageDefinedSize` and `vrStorageAllocatedSize` potentially take different values only if `vrStorageType` is `sparse`.



### 3. Definitions

This section defines two MIB modules; VMM-MIB and VIRTUAL-RESOURCES-TYPES. VMM-MIB contains the objects related to system and software information of a hypervisor, the list of virtual machines running on the hypervisor, and virtual resources allocated by the hypervisor to virtual machines. VIRTUAL-RESOURCES-TYPES contains the types of virtual resources referred from vrDeviceType in VMM-MIB. VIRTUAL-RESOURCES-TYPES might be frequently changed due to support of device types by hypervisor implementations. The following is the definitions of these modules.

```
VMM-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE, TimeTicks,
    Counter32, Counter64, Integer32, mib-2
        FROM SNMPv2-SMI
    OBJECT-GROUP, MODULE-COMPLIANCE, NOTIFICATION-GROUP
        FROM SNMPv2-CONF
    DisplayString, TEXTUAL-CONVENTION, PhysAddress, AutonomousType
        FROM SNMPv2-TC
    InterfaceIndexOrZero
        FROM IF-MIB;
```

```
vmmMibModule MODULE-IDENTITY
```

```
    LAST-UPDATED "201210200000Z"          -- 20 October 2012
    ORGANIZATION "IETF Operations and Management Area Working Group"
    CONTACT-INFO
        "      Hirochika Asai
          The University of Tokyo
          7-3-1 Hongo
          Bunkyo-ku, Tokyo 113-8656
          Japan

          +81 3 5841 6748
          panda@hongo.wide.ad.jp"
```

```
DESCRIPTION
```

```
    "This MIB is for use in managing virtual machines on a
    hypervisor. The OID `23456' is temporary one, and it
    must be assigned by IANA when this becomes an official
    document."
```

```
REVISION "201210200000Z"          -- 20 October 2012
```

```
DESCRIPTION
```

```
    "The original version of this MIB, published as
    RFCXXXX."
```

```
::= { vmmMIBAdminInfo 1 }
```



vmm-mib            OBJECT IDENTIFIER ::= { mib-2 23456 }

hypervisor        OBJECT IDENTIFIER ::= { vmm-mib 1 }

vm                OBJECT IDENTIFIER ::= { vmm-mib 2 }

vr                OBJECT IDENTIFIER ::= { vmm-mib 3 }

vmmMIBAdminInfo OBJECT IDENTIFIER ::= { vmm-mib 4 }

VirtualMachineIndex ::= TEXTUAL-CONVENTION

    DISPLAY-HINT "d"

    STATUS        current

    DESCRIPTION

        "A unique value, greater than zero, for each virtual machine in the managed hypervisor. The value for each virtual machine must remain constant at least from one re-initialization of the entity's hypervisor to the next re-initialization."

    SYNTAX        Integer32 (1..2147483647)

VirtualMachineUUID ::= TEXTUAL-CONVENTION

    DISPLAY-HINT "4x-2x-2x-2x-6x"

    STATUS        current

    DESCRIPTION

        "A unique value, a 128-bit value guaranteed to be unique over both space and time represented as a hyphen-punctuated ASCII string of the form `4x-2x-2x-2x-6x', for each virtual machine in the managed hypervisor. See [[RFC4122](#)]."

    SYNTAX        OCTET STRING (SIZE (16))

VrDeviceIndex     ::= TEXTUAL-CONVENTION

    DISPLAY-HINT "d"

    STATUS        current

    DESCRIPTION

        "A unique value, greater than zero, for each virtual device contained by the hypervisor."

    SYNTAX        Integer32 (1..2147483647)

VmVrDeviceIndex   ::= TEXTUAL-CONVENTION

    DISPLAY-HINT "d"

    STATUS        current

    DESCRIPTION

        "A unique value, greater than zero, for each virtual device contained by a virtual machine. For the indexes, sequential values are usually used."

    SYNTAX        Integer32 (1..2147483647)

-- The hypervisor group

--



-- A collection of objects common to all hypervisors.

--

hvSoftware OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..255))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A textual description of the hypervisor software. This value should not include its version, and it should be included in `hvVersion`."

::= { hypervisor 1 }

hvVersion OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..255))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A textual description of the version of the hypervisor software."

::= { hypervisor 2 }

hvObjectID OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The vendor's authoritative identification of the hypervisor software contained in the entity. This value is allocated within the SMI enterprises subtree (1.3.6.1.4.1). Note that this is different from sysObjectID in the SNMPv2-MIB [[RFC3418](#)] because sysObjectID is not the identification of the hypervisor software but the device, firmware, or management operating system."

::= { hypervisor 3 }

hvUpTime OBJECT-TYPE

SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The time (in centi-seconds) since the hypervisor was last re-initialized. Note that this is different from sysUpTime in the SNMPv2-MIB [[RFC3418](#)] and hrSystemUptime in the HOST-RESOURCES-MIB [[RFC2790](#)] because sysUpTime is the uptime of the network management portion of the system, and hrSystemUptime is the uptime of the management operating system but not the hypervisor





```
        software."
 ::= { hypervisor 4 }

-- The virtual machine group
--
-- A collection of objects common to all virtual machines.
--
vmNumber OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of virtual machines (regardless of their
        current state) present on this hypervisor."
    ::= { vm 1 }

vmTableLastChange OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The value of sysUpTime at the time of the last creation
        or deletion of an entry in the vmTable."
    ::= { vm 2 }

vmTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF VmEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "A list of virtual machine entries. The number of
        entries is given by the value of vmNumber."
    ::= { vm 3 }

vmEntry OBJECT-TYPE
    SYNTAX      VmEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An entry containing management information applicable
        to a particular virtual machine."
    INDEX      { vmIndex }
    ::= { vmTable 1 }

VmEntry ::=
    SEQUENCE {
        vmIndex          VirtualMachineIndex,
        vmName            DisplayString,
```



vmUUID	VirtualMachineUUID,
vmOSType	DisplayString,
vmAdminState	INTEGER,
vmState	INTEGER,
vmCpuTime	Counter64,
vmAutoStart	INTEGER,
vmPersistent	INTEGER

}

**vmIndex OBJECT-TYPE**

SYNTAX VirtualMachineIndex

MAX-ACCESS read-only

STATUS current

**DESCRIPTION**

"A unique value, greater than zero, for each virtual machine. It is recommended that values are assigned contiguously starting from 1. The value for each virtual machine must remain constant at least from one re-initialization of the entity's hypervisor to the next re-initialization."

::= { vmEntry 1 }

**vmName OBJECT-TYPE**

SYNTAX DisplayString (SIZE (0..255))

MAX-ACCESS read-only

STATUS current

**DESCRIPTION**

"A textual string containing information about the virtual machine name."

::= { vmEntry 2 }

**vmUUID OBJECT-TYPE**

SYNTAX VirtualMachineUUID

MAX-ACCESS read-only

STATUS current

**DESCRIPTION**

"A textual hyphen-punctuated ASCII string of the virtual machine's 128-bit UUID."

::= { vmEntry 3 }

**vmOSType OBJECT-TYPE**

SYNTAX DisplayString (SIZE (0..255))

MAX-ACCESS read-only

STATUS current

**DESCRIPTION**

"A textual string containing operating system information running on the virtual machine."

::= { vmEntry 4 }



## vmAdminState OBJECT-TYPE

```
SYNTAX      INTEGER {
                unknown(0),    -- unknown
                on(1),         -- power on
                off(2),        -- power off
                pause(3)       -- hibernate / suspend
            }
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "The administrative power state of the virtual machine.
    Note that a virtual machine is supposed to be resumed
    when vmAdminState of the virtual machine is changed from
    pause(3) to on(1)."
```

::= { vmEntry 5 }

## vmState OBJECT-TYPE

```
SYNTAX      INTEGER {
                unknown(0),    -- unknown state
                noState(1),    -- no state
                running(2),    -- running
                blocked(3),    -- blocked on resource
                paused(4),     -- paused by user
                shutdown(5),   -- being shutdown
                shutoff(6),    -- shutoff
                crashed(7)     -- crashed
            }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The current state of the virtual machine."
```

::= { vmEntry 6 }

## vmCpuTime OBJECT-TYPE

```
SYNTAX      Counter64
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The total CPU utilization time in nanosecond.  If the
    number of virtual CPUs is larger than 1, vmCpuTime may
    exceed real time."
```

::= { vmEntry 7 }

## vmAutoStart OBJECT-TYPE

```
SYNTAX      INTEGER {
                unknown(0),    -- unknown
                enable(1),     -- enabled
                disable(2)
            }
```



```

    }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "The autostart configuration of the virtual machine."
    ::= { vmEntry 8 }

```

```

vmPersistent OBJECT-TYPE
    SYNTAX INTEGER {
        unknown(0),      -- unknown
        persistent(1),   -- persistent
        transient(2)     -- transient
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This value indicates whether the virtual machine has a
        persistent configuration which means the virtual machine
        will still exist after shutting down."
    ::= { vmEntry 9 }

```

-- The virtual devices on each virtual machines

```

vmDeviceTable OBJECT-TYPE
    SYNTAX SEQUENCE OF VmDeviceEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "A list of virtual device entries corresponding to
        virtual machines contained by the hypervisor."
    ::= { vm 4 }

```

```

vmDeviceEntry OBJECT-TYPE
    SYNTAX VmDeviceEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry containing management information applicable
        to a particular virtual machine device."
    INDEX { vmIndex, vrDeviceIndex }
    ::= { vmDeviceTable 1 }

```

```

VmDeviceEntry ::=
    SEQUENCE {
        vmVrDeviceIndex          VmVrDeviceIndex
    }

```

```

vmVrDeviceIndex OBJECT-TYPE

```





```
SYNTAX      VmVrDeviceIndex
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "A unique value, greater than zero, for each virtual
    device contained by the virtual machine. The value for
    each virtual device must remain constant at least from
    one re-initialization of the virtual machine to the next
    re-initialization."
 ::= { vmDeviceEntry 1 }

-- Conformance
vmConformance OBJECT IDENTIFIER ::= { vm 5 }
vmGroups      OBJECT IDENTIFIER ::= { vmConformance 1 }
vmCompliances OBJECT IDENTIFIER ::= { vmConformance 2 }

-- Compliance statement
vmCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
        "The compliance statement for SNMP entities which have
        virtual machines."
    MODULE
        MANDATORY-GROUPS { vmNotificationGroup }
    ::= { vmCompliances 1 }

vmNotificationGroup NOTIFICATION-GROUP
    NOTIFICATIONS {
        vmAdminStateChange
    }
    STATUS      current
    DESCRIPTION
        "The notifications which indicate specific changes in the
        value of vmAdminState."
    ::= { vmGroups 1 }

-- Trap
vmTrap OBJECT IDENTIFIER ::= { vm 6 }

vmAdminStateChange NOTIFICATION-TYPE
    OBJECTS { vmIndex, vmName, vmUUID, vmAdminState, vmState }
    STATUS      current
    DESCRIPTION
        "A vmAdminStateChange trap signifies that the SNMP
        entity, acting in an agent role, has detected the
        changes in the value of vmAdminState object."
    ::= { vmTrap 1 }
```



```
-- The virtual resources group
--
-- A collection of objects common to all virtual resources.
-- This document defines some overlapped objects with hrStorage in
-- HOST-RESOURCES-MIB [RFC2790], because virtual resources shall be
-- allocated from the hypervisor's resources, which is the 'host
-- resources'.
--
vrDevice          OBJECT IDENTIFIER ::= { vr 1 }

-- defined in VIRTUAL-RESOURCE-TYPES
vrDeviceTypes     OBJECT IDENTIFIER ::= { vrDevice 1 }

vrDeviceTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF VrDeviceEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "A conceptual table of virtual devices contained by the
        hypervisor."
    ::= { vrDevice 2 }

vrDeviceEntry OBJECT-TYPE
    SYNTAX          VrDeviceEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "A conceptual entry for a virtual device contained by
        the hypervisor."
    INDEX { vrDeviceIndex }
    ::= { vrDeviceTable 1 }

VrDeviceEntry ::=
    SEQUENCE {
        vrDeviceIndex          VrDeviceIndex,
        vrParentDeviceIndex    Integer32,
        vrDeviceType           AutonomousType,
        vrDeviceDescr          DisplayString
    }

vrDeviceIndex OBJECT-TYPE
    SYNTAX          VrDeviceIndex
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "A unique value, greater than zero, for each virtual
        device contained by the hypervisor. The value for each
        virtual device must remain constant at least from one
```



re-initialization of the agent to the next  
re-initialization."  
::= { vrDeviceEntry 1 }

vrParentDeviceIndex OBJECT-TYPE

SYNTAX Integer32 (1..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of hrDeviceIndex which is the parent (i.e.,  
physical) device of this virtual device. If this  
virtual device is not any child represented in the  
hrDeviceTable, then this value shall be zero."

::= { vrDeviceEntry 2 }

vrDeviceType OBJECT-TYPE

SYNTAX AutonomousType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"An indication of the type of device.

If this value is

`vrDeviceProcessor { vrDeviceTypes 3 }' then an entry  
exists in the vrProcessorTable which corresponds to this  
device.

If this value is

`vrDeviceMemory { vrDeviceTypes 4 }' then an entry  
exists in the vrMemoryTable which corresponds to this  
device.

If this value is

`vrDeviceNetwork { vrDeviceTypes 5 }' then an entry  
exists in the vrNetworkTable which corresponds to this  
device.

If this value is

`vrDeviceStorage { vrDeviceTypes 6 }' then an entry  
exists in the vrStorageTable which corresponds to this  
device."

::= { vrDeviceEntry 3 }

vrDeviceDescr OBJECT-TYPE

SYNTAX DisplayString (SIZE(0..255))

MAX-ACCESS read-only

STATUS current

DESCRIPTION



```
        "A textual description of this virtual device, including
        the device manufacturer and revision."
 ::= { vrDeviceEntry 4 }

-- The virtual processor group
vrProcessor OBJECT IDENTIFIER ::= { vrDevice 3 }

vrProcessorTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF VrProcessorEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The table of virtual CPUs provided by the hypervisor.

        Note that this table is potentially sparse: an entry
        exists only if the correspondent value of the
        vrDeviceType is `vrDeviceProcessor'."
 ::= { vrProcessor 1 }

vrProcessorEntry OBJECT-TYPE
    SYNTAX      VrProcessorEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry for one virtual processor provided by the
        hypervisor. The vrDeviceIndex in the index represents
        the entry in the vrDeviceTable that corresponds to the
        vrProcessorEntry."
    INDEX { vrDeviceIndex }
 ::= { vrProcessorTable 1 }

VrProcessorEntry ::=
    SEQUENCE {
        vrCpuTime          Counter64
    }

vrCpuTime OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The total CPU utilization time of this virtual
        processor in nanosecond."
 ::= { vrProcessorEntry 1 }

vrProcessorAffinityTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF VrProcessorAffinityEntry
    MAX-ACCESS  not-accessible
```





STATUS current  
DESCRIPTION  
"The table of affinity of a virtual CPU provided by the  
hypervisor.  
  
Note that this table is potentially sparse: a set of  
entries exist only if the correspondent value of the  
vrDeviceType is `vrDeviceProcessor'."  
::= { vrProcessor 2 }

vrProcessorAffinityEntry OBJECT-TYPE  
SYNTAX VrProcessorAffinityEntry  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION  
"An entry containing CPU affinity associated with a  
particular virtual machine."  
INDEX { vrDeviceIndex, vrPhysicalCpuIndex }  
::= { vrProcessorAffinityTable 1 }

VrProcessorAffinityEntry ::=

```
SEQUENCE {
    vrPhysicalCpuIndex      Integer32,
    vrCpuAffinity           INTEGER,
    vrPhysicalCpuTime       Counter64
}
```

vrPhysicalCpuIndex OBJECT-TYPE  
SYNTAX Integer32 (1..2147483647)  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The value of hrDeviceIndex which corresponds to this  
virtual processor. Note that this device must be  
represented in the vrProcessorTable."  
::= { vrProcessorAffinityEntry 1 }

vrCpuAffinity OBJECT-TYPE  
SYNTAX INTEGER {  
 unknown(0), -- unknown  
 enable(1), -- enabled  
 diable(2) -- disabled  
}  
MAX-ACCESS read-write  
STATUS current  
DESCRIPTION  
"The CPU affinity to the physical CPU represented by  
vrPhysicalCpuIndex of this virtual CPU."



```
 ::= { vrProcessorAffinityEntry 2 }
```

vrPhysicalCpuTime OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The CPU utilization time of this virtual CPU corresponding to this hypervisor's CPU in nanosecond. This shall be zero if the hypervisor does not maintain such information."

```
 ::= { vrProcessorAffinityEntry 3 }
```

vrMemoryTable OBJECT-TYPE

SYNTAX SEQUENCE OF VrMemoryEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The table of virtual memory modules provided by the hypervisor.

Note that this table is potentially sparse: an entry exists only if the correspondent value of the vrDeviceType is `vrDeviceMemory'."

```
 ::= { vrDevice 4 }
```

vrMemoryEntry OBJECT-TYPE

SYNTAX VrMemoryEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry for one virtual memory module provided by the hypervisor. The vrDeviceIndex in the index represents the entry in the vrDeviceTable that corresponds to the vrMemoryEntry."

INDEX { vrDeviceIndex }

```
 ::= { vrMemoryTable 1 }
```

VrMemoryEntry ::=

SEQUENCE {

vrMemUnit	Integer32,
vrMemMax	Integer32,
vrMemMin	Integer32,
vrMemCur	Integer32

}

vrMemUnit OBJECT-TYPE

SYNTAX Integer32 (1..2147483647)



MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "The multiplication unit for vrMemMax, vrMemMin, and  
    vrMemCur. For example, when this value is 4096, the  
    memory size unit for vrMemMax, vrMem, and vrMemCur is  
    KiB."  
::= { vrMemoryEntry 1 }

## vrMemMax OBJECT-TYPE

SYNTAX Integer32 (0..2147483647)  
MAX-ACCESS read-write  
STATUS current  
DESCRIPTION  
    "The maximum memory size defined to the virtual machine  
    in the unit designated by vrMemUnit."  
::= { vrMemoryEntry 2 }

## vrMemMin OBJECT-TYPE

SYNTAX Integer32 (0..2147483647)  
MAX-ACCESS read-write  
STATUS current  
DESCRIPTION  
    "The minimum memory size defined to the virtual machine  
    in the unit designated by vrMemUnit."  
::= { vrMemoryEntry 3 }

## vrMemCur OBJECT-TYPE

SYNTAX Integer32 (0..2147483647)  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "The current memory size allocated to the virtual  
    memory module in the unit designated by vrMemUnit."  
::= { vrMemoryEntry 4 }

## vrNetworkTable OBJECT-TYPE

SYNTAX SEQUENCE OF VrNetworkEntry  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION  
    "The table of virtual network interfaces provided by the  
    hypervisor.  
  
    Note that this table is potentially sparse: an entry  
    exists only if the correspondent value of the  
    vrDeviceType is `vrDeviceNetwork'."  
::= { vrDevice 5 }



**vrNetworkEntry OBJECT-TYPE**

SYNTAX VrNetworkEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"An entry for one virtual network interface provided by the hypervisor. The vrDeviceIndex in the index represents the entry in the vrDeviceTable that corresponds to the vrNetworkEntry."

INDEX { vrDeviceIndex }

::= { vrNetworkTable 1 }

**VrNetworkEntry ::=**

SEQUENCE {

vrNetworkIndex InterfaceIndexOrZero,

vrNetworkModel DisplayString,

vrNetworkPhysAddress PhysAddress

}

**vrNetworkIndex OBJECT-TYPE**

SYNTAX InterfaceIndexOrZero

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The value of ifIndex which corresponds to this virtual network interface. If this device is not represented in the ifTable, then this value shall be zero. The corresponding ifIndex must be the virtual interface associated with an interface of the virtual machine at the hypervisor."

::= { vrNetworkEntry 1 }

**vrNetworkModel OBJECT-TYPE**

SYNTAX DisplayString (SIZE (0..255))

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"A textual string containing the (emulated) model of virtual network interface."

::= { vrNetworkEntry 2 }

**vrNetworkPhysAddress OBJECT-TYPE**

SYNTAX PhysAddress

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The MAC address of virtual network interface."

::= { vrNetworkEntry 3 }





## vrStorageTable OBJECT-TYPE

SYNTAX SEQUENCE OF VrStorageEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"The table of virtual storage provided by the hypervisor.

Note that this table is potentially sparse: an entry exists only if the correspondent value of the vrDeviceType is `vrDeviceStorage'."

::= { vrDevice 6 }

## vrStorageEntry OBJECT-TYPE

SYNTAX VrStorageEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"An entry for one virtual storage provided by the hypervisor. The vrDeviceIndex in the index represents the entry in the vrDeviceTable that corresponds to the vrStorageEntry."

INDEX { vrDeviceIndex }

::= { vrStorageTable 1 }

## VrStorageEntry ::=

## SEQUENCE {

vrStorageType	INTEGER,
vrStorageTypeHint	DisplayString,
vrStorageResourceID	DisplayString,
vrStorageAccess	INTEGER,
vrStorageMedia	INTEGER,
vrStorageMediaHint	DisplayString,
vrStorageSizeUnit	Integer32,
vrStorageDefinedSize	Integer32,
vrStorageAllocatedSize	Integer32

}

## vrStorageType OBJECT-TYPE

SYNTAX INTEGER {

other(1),	-- other format
unknown(2),	-- unknown format
block(3),	-- block device
raw(4),	-- raw file
sparse(5),	-- sparse file
network(6)	-- network

}

MAX-ACCESS read-only



```
STATUS      current
DESCRIPTION
    "The type of the virtual storage."
 ::= { vrStorageEntry 1 }

vrStorageTypeHint OBJECT-TYPE
SYNTAX      DisplayString (SIZE (0..255))
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "A textual string of the virtual storage type.  For
     example, this represents the specific format name of the
     sparse file."
 ::= { vrStorageEntry 2 }

vrStorageResourceID OBJECT-TYPE
SYNTAX      DisplayString (SIZE (0..255))
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "A textual string that represents the resource
     identifier of the virtual storage.  For example, this
     contains the path to the disk image file that
     corresponds to the virtual storage."
 ::= { vrStorageEntry 3 }

vrStorageAccess OBJECT-TYPE
SYNTAX      INTEGER {
                    readwrite(1), -- read-write
                    readonly(2)  -- read-only
                    }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The access permission of the virtual storage."
 ::= { vrStorageEntry 4 }

vrStorageMedia OBJECT-TYPE
SYNTAX      INTEGER {
                    other(1),          -- other media
                    unknown(2),       -- unknown media
                    hardDisk(3),      -- e.g., IDE and SCSI
                    floppyDisk(4),
                    opticalDiskROM(5),
                    opticalDiskWOMR(6), -- Write Once Read Many
                    opticalDiskRW(7)
                    }
MAX-ACCESS  read-only
```



STATUS       current  
DESCRIPTION  
    "The media type of the virtual storage."  
::= { vrStorageEntry 5 }

vrStorageMediaHint OBJECT-TYPE

SYNTAX       DisplayString (SIZE (0..255))  
MAX-ACCESS   read-only  
STATUS       current  
DESCRIPTION  
    "A textual string of the virtual storage media. For  
    example, this represents the specific driver name of the  
    media such as IDE and SCSI."  
::= { vrStorageEntry 6 }

vrStorageSizeUnit OBJECT-TYPE

SYNTAX       Integer32 (1..2147483647)  
MAX-ACCESS   read-only  
STATUS       current  
DESCRIPTION  
    "The multiplication unit for vrStorageDefinedSize and  
    vrStorageAllocatedSize. For example, when this value is  
    1048576, the storage size unit for vrStorageDefinedSize  
    and vrStorageAllocatedSize is MiB."  
::= { vrStorageEntry 7 }

vrStorageDefinedSize OBJECT-TYPE

SYNTAX       Integer32 (0..2147483647)  
MAX-ACCESS   read-only  
STATUS       current  
DESCRIPTION  
    "The defined virtual storage size defined in the unit  
    designated by vrStorageSizeUnit. If this information is  
    not available, this value shall be zero."  
::= { vrStorageEntry 8 }

vrStorageAllocatedSize OBJECT-TYPE

SYNTAX       Integer32 (0..2147483647)  
MAX-ACCESS   read-only  
STATUS       current  
DESCRIPTION  
    "The storage size allocated to the virtual storage from  
    a physical storage in the unit designated by  
    vrStorageSizeUnit. When the virtual storage is block  
    device or raw file, this value and vrStorageDefinedSize  
    are supposed to equal. This value is potentially  
    different from vrStorageDefinedSize only if the  
    vrStorageType is `sparse'. If this information is not



```
        available, this value shall be zero."
 ::= { vrStorageEntry 9 }
```

END

VIRTUAL-RESOURCES-TYPES DEFINITIONS ::= BEGIN

IMPORTS

```
    MODULE-IDENTITY, OBJECT-IDENTITY
        FROM SNMPv2-SMI
    vmmMIBAdminInfo, vrDevice
        FROM VMM-MIB;
```

vrTypesModule MODULE-IDENTITY

```
    LAST-UPDATED "201210200000Z"          -- 20 October 2012
    ORGANIZATION "IETF Operations and Management Area Working Group"
    CONTACT-INFO
        "      Hirochika Asai
           The University of Tokyo
           7-3-1 Hongo
           Bunkyo-ku, Tokyo 113-8656
           Japan

           +81 3 5841 6748
           panda@hongo.wide.ad.jp"
```

DESCRIPTION

```
"This MIB module registers type definitions for virtual
resource device types.
```

```
After the initial revision, this module will be
maintained by IANA."
```

```
REVISION "201210200000Z"          -- 20 October 2012
```

DESCRIPTION

```
"The original version of this MIB, published as
RFCXXXX."
```

```
::= { vmmMIBAdminInfo 2 }
```

vrDeviceTypes OBJECT IDENTIFIER ::= { vrDevice 1 }

vrDeviceOther OBJECT-IDENTITY

STATUS current

DESCRIPTION

```
"The device type identifier used when no other defined
type is appropriate."
```

```
::= { vrDeviceTypes 1 }
```

vrDeviceUnknown OBJECT-IDENTITY

STATUS current





```
DESCRIPTION
    "The device type identifier used when the device type is
    unknown."
::= { vrDeviceTypes 2 }

vrDeviceProcessor OBJECT-IDENTITY
    STATUS current
    DESCRIPTION
        "The device type identifier used for a CPU."
    ::= { vrDeviceTypes 3 }

vrDeviceMemory OBJECT-IDENTITY
    STATUS current
    DESCRIPTION
        "The device type identifier used for a memory module."
    ::= { vrDeviceTypes 4 }

vrDeviceNetwork OBJECT-IDENTITY
    STATUS current
    DESCRIPTION
        "The device type identifier used for a network
        interface."
    ::= { vrDeviceTypes 5 }

vrDeviceStorage OBJECT-IDENTITY
    STATUS current
    DESCRIPTION
        "The device type identifier used for a storage device such
        as a disk drive."
    ::= { vrDeviceTypes 6 }

END
```



#### **4. IANA Considerations**

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER values recorded in the SMI Numbers registry:

Descriptor -----	OBJECT IDENTIFIER value -----
vmm-mib	{ mib-2 TBD }

## 5. Security Considerations

There are a number of management objects defined in this MIB that have a MAX-ACCESS clause of read-write and/or 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 hypervisor and virtual machine operations.

There are a number of managed objects in this MIB that may contain sensitive information. The objects in the hvSoftware and hvVersion list information about the hypervisor's software and version. Some may wish not to disclose to others which software they are running. Further, an inventory of the running software and versions may be helpful to an attacker who hopes to exploit software bugs in certain applications. Moreover, the objects in the vmTable and vrDeviceTable list information about the virtual machines, and their resources. Some may wish not to disclose to others how many and what virtual machines they are operating.

It is thus important to control even GET access to these objects and possibly to even encrypt the values of these object when sending them over the network via SNMP. Not all versions of SNMP provide features for such a secure environment.

It is recommended that attention be specifically given to implementing the MAX-ACCESS clause in a number of objects, including vmAdminState, vmAutoStart, vrMemMax, vrMemMin, and vrProcessorAffinity in scenarios that DO NOT use SNMPv3 strong security (i.e. authentication and encryption). Extreme caution must be used to minimize the risk of cascading security vulnerabilities when SNMPv3 strong security is not used. When SNMPv3 strong security is not used, these objects should have access of read-only, not read-create.

SNMPv1 by itself is not a secure environment. Even if the network itself is secure (for example by using IPsec), even then, 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.

It is recommended that the implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model [RFC 3414](#) [[RFC3414](#)] and the View-based Access Control Model [RFC 3415](#) [[RFC3415](#)] is recommended.

It is then a customer/user responsibility to ensure that the SNMP entity giving access to an instance of this MIB, 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.

## 6. Normative References

- [RFC1155] Rose, M. and K. McCloghrie, "Structure and identification of management information for TCP/IP-based internets", STD 16, [RFC 1155](#), May 1990.
- [RFC1157] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol (SNMP)", STD 15, [RFC 1157](#), May 1990.
- [RFC1212] Rose, M. and K. McCloghrie, "Concise MIB definitions", STD 16, [RFC 1212](#), March 1991.
- [RFC1215] Rose, M., "Convention for defining traps for use with the SNMP", [RFC 1215](#), March 1991.
- [RFC1901] Case, J., McCloghrie, K., McCloghrie, K., Rose, M., and S. Waldbusser, "Introduction to Community-based SNMPv2", [RFC 1901](#), January 1996.
- [RFC2573] Levi, D., Meyer, P., and B. Stewart, "SNMP Applications", [RFC 2573](#), April 1999.
- [RFC2575] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", [RFC 2575](#), April 1999.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, [RFC 2578](#), April 1999.
- [RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "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.
- [RFC2790] Waldbusser, S. and P. Grillo, "Host Resources MIB", [RFC 2790](#), March 2000.
- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", [RFC 2863](#), June 2000.
- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", [RFC 3410](#), December 2002.





- [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.
- [RFC3412] Case, J., Harrington, D., Presuhn, R., and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", STD 62, [RFC 3412](#), December 2002.
- [RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, [RFC 3414](#), December 2002.
- [RFC3415] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", STD 62, [RFC 3415](#), December 2002.
- [RFC3416] Presuhn, R., "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)", STD 62, [RFC 3416](#), December 2002.
- [RFC3417] Presuhn, R., "Transport Mappings for the Simple Network Management Protocol (SNMP)", STD 62, [RFC 3417](#), December 2002.
- [RFC3418] Presuhn, R., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, [RFC 3418](#), December 2002.
- [RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", [RFC 4122](#), July 2005.



## Authors' Addresses

Hirochika Asai  
The University of Tokyo  
7-3-1 Hongo  
Bunkyo-ku, Tokyo 113-8656  
JP

Phone: +81 3 5841 6748  
Email: panda@hongo.wide.ad.jp

Yuji Sekiya  
The University of Tokyo  
2-11-16 Yayoi  
Bunkyo-ku, Tokyo 113-8658  
JP

Email: sekiya@wide.ad.jp

Keiichi Shima  
IIJ Innovation Institute Inc.  
1-105 Kanda-Jinbocho  
Chiyoda-ku, Tokyo 101-0051  
JP

Email: keiichi@iijlab.net

Hiroshi Esaki  
The University of Tokyo  
7-3-1 Hongo  
Bunkyo-ku, Tokyo 113-8656  
JP

Phone: +81 3 5841 6748  
Email: hiroshi@wide.ad.jp

