

OPSAWG
Internet-Draft
Intended status: Standards Track
Expires: August 29, 2013

H. Asai
Univ. of Tokyo
M. MacFaden
VMware Inc.
J. Schoenwaelder
Jacobs University
Y. Sekiya
Univ. of Tokyo
K. Shima
IIJ Innovation Institute Inc.
T. Tsou
Huawei Technologies (USA)
C. Zhou
Huawei Technologies
H. Esaki
Univ. of Tokyo
February 25, 2013

**Management Information Base for Virtual Machines Controlled by a
Hypervisor
draft-asai-vmm-mib-02**

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 objects for managing virtual machines controlled by a hypervisor (a.k.a. virtual machine manager).

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 August 29, 2013.

Copyright Notice

Copyright (c) 2013 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

1.	Introduction	3
1.1.	Requirements Language	3
2.	The Internet-Standard Management Framework	4
3.	Managed Objects for Virtual Machines Controlled by a Hypervisor	5
3.1.	Managed Objects on Virtualization Environment	5
3.2.	Overview of the MIB Module	6
3.3.	Definitions	8
4.	IANA Considerations	34
5.	Security Considerations	35
6.	References	37
6.1.	Normative References	37
6.2.	Informative References	37
	Authors' Addresses	38

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 objects for managing virtual machines controlled by a hypervisor (a.k.a. virtual machine managers). A hypervisor controls multiple virtual machines on a single physical machine by allocating resources to each virtual machine using virtualization technologies. Therefore, this MIB module contains information on virtual machines and their resources controlled by a hypervisor as well as hypervisor's hardware and software information.

The design of this MIB module has been derived from enterprise specific MIB modules, namely a MIB module for managing guests of the Xen hypervisor, a MIB module for managing virtual machines controlled by the VMware hypervisor, and a MIB module using the libvirt programming interface to access different hypervisors.

1.1. Requirements Language

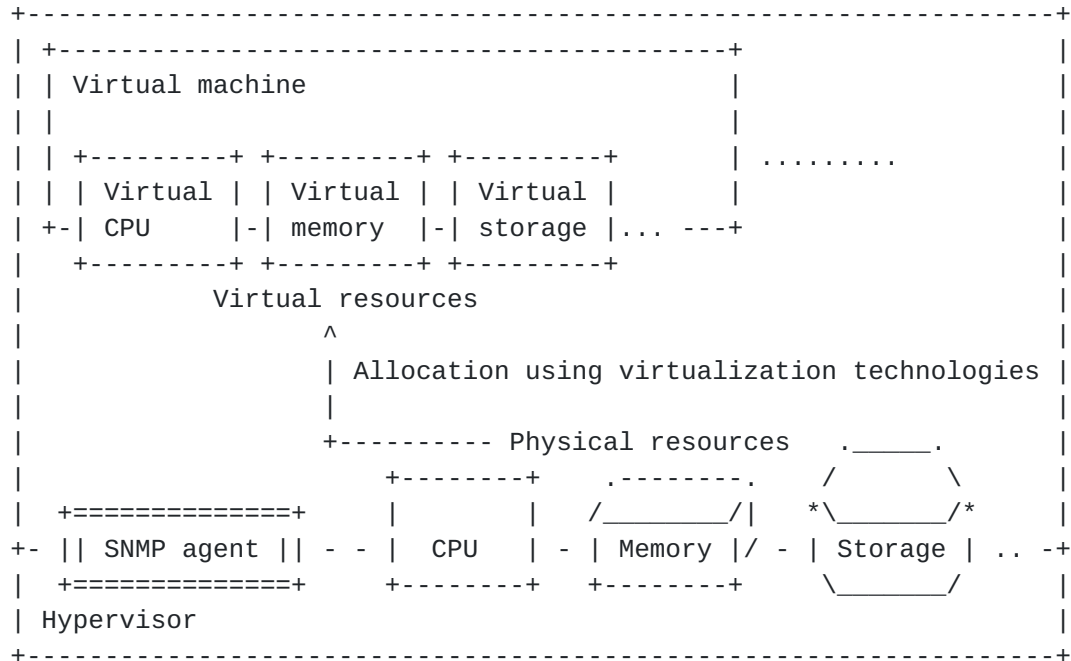
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#).

2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to [section 7 of RFC 3410](#) [[RFC3410](#)]. Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIV2, which is described in STD 58, [RFC 2578](#) [[RFC2578](#)], STD 58, [RFC 2579](#) [[RFC2579](#)] and STD 58, [RFC 2580](#) [[RFC2580](#)].

3. Managed Objects for Virtual Machines Controlled by a Hypervisor

3.1. 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 controlled by the hypervisor from physical resources.

Figure 1: An example of a virtualization environment

On the common implementations of hypervisor softwares, a hypervisor allocates resources as virtual devices such as virtual CPUs, virtual memory, virtual storage, and virtual network interface to multiple virtual machines controlled by the hypervisor from physical resources. This document defines objects related to system and software information of a hypervisor, the list of virtual machines controlled by the hypervisor, and virtual resources allocated by the hypervisor to virtual machines. As shown in Figure 1, the virtual resource objects are defined as virtual devices. Consequently, this document specifies four specific types of virtual devices; CPUs (processors), memory, network interfaces, and storage devices. Note that physical resources are managed in HOST-RESOURCES-MIB [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 [RFC2863].

The objects defined in 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; i.e., the objects do not take into account the actual resource utilization on each virtual machine but the resource allocation from the physical resources. For example, `vmNetworIfIndex` indicates the virtual interface associated with an interface of a virtual machine at the hypervisor, and consequently, the ``in'` and ``out'` directions denote ``from a virtual machine to the hypervisor'` and ``from the hypervisor to a virtual machine'`, respectively. Moreover, `vmStorageAllocatedSize` denotes the size allocated by the hypervisor, but not the size actually used by the operating system on the virtual machine. This means that `vmStorageDefinedSize` and `vmStorageAllocatedSize` must not take different values when the `vmStorageSourceType` is ``block'` or ``raw'`.

3.2. Overview of the MIB Module

The MIB module is organized into a group of scalars and tables. The scalars below ``hypervisor'` provide basic information about the hypervisor. The ``vm'` lists the virtual machines (guests) that are known to the hypervisor. The ``vmCpuTable'` and ``vmCpuAffinityTable'` provide the mapping of virtual CPUs and their affinity to virtual machines. The ``vmStorageTable'` and the ``vmNetworkTable'` provide the mapping of logical storage areas and network interfaces to virtual machines.

The ``vmOperState'` textual convention defines a state model for virtual machines. Events causing transitions between major states will cause the generation of notifications (`vmStarted`, `vmStopped`, `vmSuspended`, `vmResumed`).

The MIB module provides a few writable objects that can be used to make non-persistent changes, e.g., changing the memory allocation or the CPU allocation. It is not the goal of this MIB module to provide a configuration interface for virtual machines since other protocols and data modeling languages are more suitable for this task.

The OID tree structure of the MIB module is shown below.

```
--vmMIB (1.3.6.1.2.1.yyy)
+-vmNotifications(0)
|  +--vmStarted(1) [vmName, vmUUID, vmOperState]
|  +--vmStopped(2) [vmName, vmUUID, vmOperState]
|  +--vmSuspended(3) [vmName, vmUUID, vmOperState]
|  +--vmResumed(4) [vmName, vmUUID, vmOperState]
+-vmObjects(1)
```



```

| +vmHypervisor(1)
| |   +-- r-n SnmpAdminString    vmHvSoftware(1)
| |   +-- r-n SnmpAdminString    vmHvVersion(2)
| |   +-- r-n OBJECT IDENTIFIER  vmHvObjectID(3)
| |   +-- r-n TimeTicks          vmHvUpTime(4)
| +-- r-n Integer32    vmNumber(2)
| +-- r-n TimeTicks    vmTableLastChange(3)
| +--vmTable(4)
| |   +--vmEntry(1) [vmIndex]
| | |   +-- --- VirtualMachineIndex    vmIndex(1)
| | |   +-- r-n SnmpAdminString          vmName(2)
| | |   +-- r-n UUIDorZero              vmUUID(3)
| | |   +-- r-n SnmpAdminString          vmOSType(4)
| | |   +-- rwn VirtualMachineAdminState
| | | |                                   vmAdminState(5)
| | |   +-- r-n VirtualMachineOperState
| | | |                                   vmOperState(6)
| | |   +-- rwn VirtualMachineAutoStart
| | | |                                   vmAutoStart(7)
| | |   +-- r-n VirtualMachinePersistent
| | | |                                   vmPersistent(8)
| | |   +-- r-n Integer32                vmCurCpuNumber(9)
| | |   +-- rwn Integer32                vmMinCpuNumber(10)
| | |   +-- rwn Integer32                vmMaxCpuNumber(11)
| | |   +-- r-n Integer32                vmMemUnit(12)
| | |   +-- r-n Integer32                vmCurMem(13)
| | |   +-- rwn Integer32                vmMinMem(14)
| | |   +-- rwn Integer32                vmMaxMem(15)
| | |   +-- r-n TimeTicks                vmUpTime(16)
| | |   +-- r-n Counter64                vmCpuTime(17)
| +--vmCpuTable(5)
| |   +--vmCpuEntry(1) [vmIndex, vmCpuIndex]
| | |   +-- --- VirtualMachineCpuIndex
| | | |                                   vmCpuIndex(1)
| | |   +-- r-n Counter64                vmCpuCoreTime(2)
| +--vmCpuAffinityTable(6)
| |   +--vmCpuAffinityEntry(1) [vmIndex,
| | |   |                                   vmCpuIndex,
| | |   |                                   vmCpuPhysIndex]
| | |   +-- --- Integer32                vmCpuPhysIndex(1)
| | |   +-- rwn Integer32                vmCpuAffinity(2)
| +--vmStorageTable(7)
| |   +--vmStorageEntry(1) [vmStorageVmIndex, vmStorageIndex]
| | |   +-- --- VirtualMachineIndexOrZero
| | | |                                   vmStorageVmIndex(1)
| | |   +-- --- VirtualMachineStorageIndex
| | | |                                   vmStorageIndex(2)
| | |   +-- r-n Integer32                vmStorageParent(4)

```



```

| | +-- r-n VritualMachineStorageSourceType
| | | vmStorageSourceType(4)
| | +-- r-n SnmpAdminString vmStorageSourceTypeString(5)
| | +-- r-n SnmpAdminString vmStorageResourceID(6)
| | +-- r-n VirtualMachineStorageAccess
| | | vmStorageAccess(7)
| | +-- r-n VirtualMachineStorageMediaType
| | | vmStorageMediaType(8)
| | +-- r-n SnmpAdminString vmStorageMediaTypeString(9)
| | +-- r-n Integer32 vmStorageSizeUnit(10)
| | +-- r-n Integer32 vmStorageDefinedSize(11)
| | +-- r-n Integer32 vmStorageAllocatedSize(12)
| | +-- r-n Counter64 vmStorageReadIOs(13)
| | +-- r-n Counter64 vmStorageWriteIOs(14)
| +--vmNetworkTable(8)
|   +--vmNetworkEntry(1) [vmIndex, vmNetworkIndex]
|     +-- --- VirtualMachineNetworkIndex
|       | vmNetworIndex(1)
|       +-- r-n InterfaceIndexOrZero vmNetworIfIndex(2)
|       +-- r-n InterfaceIndexOrZero vmNetworParent(3)
|       +-- r-n SnmpAdminString vmNetworkModel(4)
|       +-- r-n PhysAddress vmNetworkPhysAddress(5)
+--vmConformance(2)
  +--vmCompliances(1)
  | +--vmFullCompliances(1)
  | +--vmReadOnlyCompliances(2)
+--vmGroups(2)
  +--vmHypervisorGroup(1)
  +--vmVirtualMachineGroup(2)
  +--vmCpuGroup(3)
  +--vmCpuAffinityGroup(4)
  +--vmStorageGroup(5)
  +--vmNetworkGroup(6)
  +--vmNotificationGroup(7)

```

3.3. Definitions

VM-MIB DEFINITIONS ::= BEGIN

IMPORTS

```

MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE, TimeTicks,
Counter64, Integer32, mib-2
FROM SNMPv2-SMI
OBJECT-GROUP, MODULE-COMPLIANCE, NOTIFICATION-GROUP
FROM SNMPv2-CONF
TEXTUAL-CONVENTION, PhysAddress
FROM SNMPv2-TC
SnmpAdminString

```


FROM SNMP-FRAMEWORK-MIB
UUIDorZero
FROM UUID-TC-MIB
InterfaceIndexOrZero
FROM IF-MIB;

vmMIB MODULE-IDENTITY

LAST-UPDATED "201302250000Z" -- 25 February 2013
ORGANIZATION "IETF Operations and Management Area Working Group"
CONTACT-INFO

"

WG E-mail: (To be added after approved by WG)
Mailing list subscription info:
http:// (To be added after approved by WG)

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

Michael MacFaden
VMware Inc.
Email: mrm@vmware.com

Juergen Schoenwaelder
Jacobs University
Campus Ring 1
Bremen 28759
Germany
Email: j.schoenwaelder@jacobs-university.de

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

"

DESCRIPTION

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

Copyright (c) 2013 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in [Section 4.c](#) of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>)."

REVISION "201302250000Z" -- 25 February 2013

DESCRIPTION

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

::= { mib-2 yyy }

vmNotifications OBJECT IDENTIFIER ::= { vmMIB 0 }
vmObjects OBJECT IDENTIFIER ::= { vmMIB 1 }
vmConformance OBJECT IDENTIFIER ::= { vmMIB 2 }

-- Textual conversion definitions

--

VirtualMachineIndex ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

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

SYNTAX Integer32 (1..2147483647)

VirtualMachineIndexOrZero ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"This textual convention is an extension of the VirtualMachineIndex convention. This extension permits the additional value of zero. The meaning of the value zero is object-specific and must therefore be defined as

part of the description of any object which uses this syntax. Examples of the usage of zero might include situations where a virtual machine is unknown, or when none or all virtual machines need to be referenced."

SYNTAX Integer32 (0..2147483647)

VirtualMachineAdminState ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The administrative state of a virtual machine:

on(1) The administrative power state of the
virtual machine is off.

off(2) The administrative power state of the
virtual machine is on.

pause(3) The administrative power state of the
virtual machine is hibernated or
suspended."

SYNTAX INTEGER {
 on(1),
 off(2),
 pause(3)
 }

VirtualMachineOperState ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The operational state of a virtual machine:

unknown(1) The state is unknown, e.g., because the
implementation failed to obtain the state
from the hypervisor.

other(2) The state has been obtained but it is
not a known state.

running(3) The virtual machine is currently running.

blocked(4) The virtual machine is currently blocked.

paused(5) The virtual machine is currently paused.

migrating(6) The virtual machine is currently
migrating.

shutdown(7) The virtual machine is currently in the

process of shutting down.

shutoff(8) The virtual machine is down.

crashed(9) The virtual machine has crashed."

```
SYNTAX          INTEGER {
                    unknown(1),
                    other(2),
                    running(3),
                    blocked(4),
                    paused(5),
                    migrating(6),
                    shutdown(7),
                    shutoff(8),
                    crashed(9)
                  }
```

VirtualMachineAutoStart ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The autostart configuration of a virtual machine:

unknown(1) The autostart configuration is unknown,
e.g., because the implementation failed
to obtain the autostart configuration
from the hypervisor. (read-only)

enable(2) The autostart configuration of the
virtual machine is enabled.

disable(3) The autostart configuration of the
virtual machine is disabled."

```
SYNTAX          INTEGER {
                    unknown(1),
                    enable(2),
                    disable(3)
                  }
```

VirtualMachinePersistent ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"This value indicates whether a virtual machine has a
persistent configuration which means the virtual machine
will still exist after shutting down:

unknown(1) The persistent configuration is unknown,
e.g., because the implementation failed
to obtain the persistent configuration

from the hypervisor. (read-only)

persistent(2) The virtual machine is persistent.

transient(3) The virtual machine is transient, i.e.,
the virtual machine does not exist after
its power-off."

```
SYNTAX      INTEGER {  
                unknown(1),  
                persistent(2),  
                transient(3)  
            }
```

VirtualMachineCpuIndex ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"A unique value, greater than zero, identifying a
virtual CPU assigned to a virtual machine. The value
for each virtual CPU must remain constant at least from
one re-initialization of the virtual machine to the next
re-initialization."

```
SYNTAX      Integer32 (1..2147483647)
```

VirtualMachineStorageIndex ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"A unique value, greater than zero, identifying a
virtual storage device allocated to a virtual machine.
The value for each virtual storage device must remain
constant at least from one re-initialization of the
virtual machine to the next re-initialization."

```
SYNTAX      Integer32 (1..2147483647)
```

VirtualMachineStorageSourceType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The source type of a virtual storage device:

unknown(1) The source type is unknown, e.g., because
the implementation failed to obtain the
media type from the hypervisor.

other(2) The source type is other than those
defined in this conversion.

block(3) The source type is a block device.

raw(4) The source type is a raw-formatted file.

sparse(5) The source type is a sparse file.

network(6) The source type is a network device."

SYNTAX INTEGER {
 unknown(1),
 other(2),
 block(3),
 raw(4),
 sparse(5),
 network(6)
 }

VirtualMachineStorageAccess ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The access permission of a virtual storage:

readwrite(1) The virtual storage is a read-write
 device.

readonly(2) The virtual storage is a read-only
 device."

SYNTAX INTEGER {
 readwrite(1),
 readonly(2)
 }

VirtualMachineStorageMediaType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The media type of a virtual storage device:

unknown(1) The media type is unknown, e.g., because
 the implementation failed to obtain the
 media type from the hypervisor.

other(2) The media type is other than those
 defined in this conversion.

hardDisk(3) The media type is hard disk.

opticalDisk(4) The media type is optical disk."

SYNTAX INTEGER {
 other(1),
 unknown(2),
 hardDisk(3),


```
        opticalDisk(4)
    }
```

VirtualMachineNetworkIndex ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"A unique value, greater than zero, identifying a virtual network interface allocated to a virtual machine. The value for each virtual network interface must remain constant at least from one re-initialization of the virtual machine to the next re-initialization."

SYNTAX Integer32 (1..2147483647)

-- The hypervisor group

--

-- A collection of objects common to all hypervisors.

--

vmHypervisor OBJECT IDENTIFIER ::= { vmObjects 1 }

vmHvSoftware OBJECT-TYPE

SYNTAX SnmpAdminString (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'."

::= { vmHypervisor 1 }

vmHvVersion OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..255))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

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

::= { vmHypervisor 2 }

vmHvObjectID 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."
 ::= { vmHypervisor 3 }

vmHvUpTime 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."
 ::= { vmHypervisor 4 }

-- The virtual machine information
--
-- A collection of objects common to all virtual machines.
--

vmNumber OBJECT-TYPE
    SYNTAX      Integer32 (0..2147483647)
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of virtual machines (regardless of their
        current state) present on this hypervisor."
 ::= { vmObjects 2 }

vmTableLastChange OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The value of hvUpTime at the time of the last creation
        or deletion of an entry in the vmTable."
 ::= { vmObjects 3 }

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."

::= { vmObjects 4 }

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	SnmpAdminString,
vmUUID	UUIDorZero,
vmOSType	SnmpAdminString,
vmAdminState	VirtualMachineAdminState,
vmOperState	VirtualMachineOperState,
vmAutoStart	VirtualMachineAutoStart,
vmPersistent	VirtualMachinePersistent,
vmCurCpuNumber	Integer32,
vmMinCpuNumber	Integer32,
vmMaxCpuNumber	Integer32,
vmMemUnit	Integer32,
vmCurMem	Integer32,
vmMinMem	Integer32,
vmMaxMem	Integer32,
vmUpTime	TimeTicks,
vmCpuTime	Counter64

}

vmIndex OBJECT-TYPE

SYNTAX VirtualMachineIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A unique value, greater than zero, identifying the virtual machine."

::= { vmEntry 1 }

vmName OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..255))

MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "A textual name of the virtual machine."
::= { vmEntry 2 }

vmUUID OBJECT-TYPE

SYNTAX UUIDorZero
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The virtual machine's 128-bit UUID or the zero-length
 string when a UUID is not available."
::= { vmEntry 3 }

vmOSType OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..255))
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "A textual description containing operating system
 information installed on the virtual machine. This
 value corresponds to the operating system the hypervisor
 assumes to be running when the virtual machine is
 started. This may differ from the actual operating
 system in case the virtual machine boots into a
 different operating system."
::= { vmEntry 4 }

vmAdminState OBJECT-TYPE

SYNTAX VirtualMachineAdminState
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 }

vmOperState OBJECT-TYPE

SYNTAX VirtualMachineOperState
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The current operational state of the virtual machine."
::= { vmEntry 6 }

vmAutoStart OBJECT-TYPE

SYNTAX VirtualMachineAutoStart
MAX-ACCESS read-write
STATUS current
DESCRIPTION
 "The autostart configuration of the virtual machine."
::= { vmEntry 7 }

vmPersistent OBJECT-TYPE

SYNTAX VirtualMachinePersistent
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 8 }

vmCurCpuNumber OBJECT-TYPE

SYNTAX Integer32 (0..2147483647)
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The number of virtual CPUs currently assigned to the
 virtual machine."
::= { vmEntry 9 }

vmMinCpuNumber OBJECT-TYPE

SYNTAX Integer32 (-1|0..2147483647)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
 "The minimum number of virtual CPUs that are assigned to
 the virtual machine when it is in a power-on state. The
 value -1 indicates that there is no hard boundary for
 the minimum number of virtual CPUs. Changes to this
 object may not persist across restarts of the
 hypervisor."
::= { vmEntry 10 }

vmMaxCpuNumber OBJECT-TYPE

SYNTAX Integer32 (-1|0..2147483647)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
 "The maximum number of virtual CPUs that are assigned to
 the virtual machine when it is in a power-on state. The
 value -1 indicates that there is no limit. Changes to

this object may not persist accross restarts of the
 hypervisor."
::= { vmEntry 11 }

vmMemUnit OBJECT-TYPE

SYNTAX Integer32 (1..2147483647)
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The multiplication unit for vmCurMem, vmMinMem, and
 vmMaxMem. For example, when this value is 1024, the
 memory size unit for vmCurMem, vmMinMem, and vmMaxMem is
 KiB."
::= { vmEntry 12 }

vmCurMem OBJECT-TYPE

SYNTAX Integer32 (0..2147483647)
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The current memory size currently allocated to the
 virtual memory module in the unit designated by
 vmMemUnit."
::= { vmEntry 13 }

vmMinMem OBJECT-TYPE

SYNTAX Integer32 (-1|0..2147483647)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
 "The minimum memory size defined to the virtual machine
 in the unit designated by vmMemUnit. The value -1
 indicates that there is no hard boundary for the minimum
 memory size. Changes to this object may not persist
 across the restart of the hypervisor."
::= { vmEntry 14 }

vmMaxMem OBJECT-TYPE

SYNTAX Integer32 (-1|0..2147483647)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
 "The maximum memory size defined to the virtual machine
 in the unit designated by vmMemUnit. The value -1
 indicates that there is no limit. Changes to this
 object may not persist across the restart of the
 hypervisor."
::= { vmEntry 15 }

vmUpTime OBJECT-TYPE

SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The time (in centi-seconds) since the administrative state of the virtual machine was last changed to power on."

::= { vmEntry 16 }

vmCpuTime OBJECT-TYPE

SYNTAX Counter64

UNITS "microsecond"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total CPU time used in microsecond. If the number of virtual CPUs is larger than 1, vmCpuTime may exceed real time."

::= { vmEntry 17 }

-- The virtual CPU on each virtual machines

vmCpuTable OBJECT-TYPE

SYNTAX SEQUENCE OF VmCpuEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The table of virtual CPUs provided by the hypervisor."

::= { vmObjects 5 }

vmCpuEntry OBJECT-TYPE

SYNTAX VmCpuEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry for one virtual processor assigned to a virtual machine."

INDEX { vmIndex, vmCpuIndex }

::= { vmCpuTable 1 }

VmCpuEntry ::=

SEQUENCE {

vmCpuIndex VirtualMachineCpuIndex,

vmCpuCoreTime Counter64

}

vmCpuIndex OBJECT-TYPE

SYNTAX VirtualMachineCpuIndex


```

MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
    "A unique value identifying a virtual CPU assigned to
    the virtual machine."
::= { vmCpuEntry 1 }

```

```

vmCpuCoreTime OBJECT-TYPE
    SYNTAX      Counter64
    UNITS        "microsecond"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The total CPU time used by this virtual CPU in
        microsecond."
    ::= { vmCpuEntry 2 }

```

-- The virtual CPU affinity on each virtual machines

```

vmCpuAffinityTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF VmCpuAffinityEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "A list of CPU affinity entries of a virtual CPU."
    ::= { vmObjects 6 }

```

```

vmCpuAffinityEntry OBJECT-TYPE
    SYNTAX      VmCpuAffinityEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An entry containing CPU affinity associated with a
        particular virtual machine."
    INDEX       { vmIndex, vmCpuIndex, vmCpuPhysIndex }
    ::= { vmCpuAffinityTable 1 }

```

```

VmCpuAffinityEntry ::=
    SEQUENCE {
        vmCpuPhysIndex      Integer32,
        vmCpuAffinity        Integer32
    }

```

```

vmCpuPhysIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "A value identifying a physical CPU on the hypervisor."

```


On systems implementing the HOST-RESOURCES-MIB, the value must be the same value that is used as the index in the hrProcessorTable (hrDeviceIndex)."

```
::= { vmCpuAffinityEntry 2 }
```

vmCpuAffinity OBJECT-TYPE

```
SYNTAX      INTEGER {
                unknown(0),    -- unknown
                enable(1),     -- enabled
                disable(2)     -- disabled
            }
MAX-ACCESS   read-write
STATUS       current
DESCRIPTION
    "The CPU affinity of this virtual CPU to the physical
    CPU represented by `vmCpuPhysIndex'."
::= { vmCpuAffinityEntry 3 }
```

```
-- The virtual storage devices on each virtual machine.  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'
```

vmStorageTable OBJECT-TYPE

```
SYNTAX      SEQUENCE OF VmStorageEntry
MAX-ACCESS   not-accessible
STATUS       current
DESCRIPTION
    "The conceptual table of virtual storage devices
    attached to the virtual machine."
::= { vmObjects 7 }
```

vmStorageEntry OBJECT-TYPE

```
SYNTAX      VmStorageEntry
MAX-ACCESS   not-accessible
STATUS       current
DESCRIPTION
    "An entry for one virtual storage device attached to the
    virtual machine."
INDEX { vmStorageVmIndex, vmStorageIndex }
::= { vmStorageTable 1 }
```

VmStorageEntry ::=

```
SEQUENCE {
    vmStorageVmIndex      VirtualMachineIndexOrZero,
    vmStorageIndex        VirtualMachineStorageIndex,
    vmStorageParent        Integer32,
```



```

    vmStorageSourceType      VirtualMachineStorageSourceType,
    vmStorageSourceTypeString
                                SnmpAdminString,
    vmStorageResourceID      SnmpAdminString,
    vmStorageAccess          VirtualMachineStorageAccess,
    vmStorageMediaType       VirtualMachineStorageMediaType,
    vmStorageMediaTypeString
                                SnmpAdminString,
    vmStorageSizeUnit        Integer32,
    vmStorageDefinedSize     Integer32,
    vmStorageAllocatedSize   Integer32,
    vmStorageReadIOs         Counter64,
    vmStorageWriteIOs        Counter64
}

```

vmStorageVmIndex OBJECT-TYPE

```

SYNTAX      VirtualMachineIndexOrZero
MAX-ACCESS  not-accessible
STATUS      current

```

DESCRIPTION

"This value identifies the virtual machine (guest) this storage device has been allocated to. The value zero indicates that the storage device is currently not allocated to any virtual machines."

```
 ::= { vmStorageEntry 1 }
```

vmStorageIndex OBJECT-TYPE

```

SYNTAX      VirtualMachineStorageIndex
MAX-ACCESS  not-accessible
STATUS      current

```

DESCRIPTION

"A unique value identifying a virtual storage device allocated to the virtual machine."

```
 ::= { vmStorageEntry 2 }
```

vmStorageParent OBJECT-TYPE

```

SYNTAX      Integer32 (0..2147483647)
MAX-ACCESS  read-only
STATUS      current

```

DESCRIPTION

"The value of hrStorageIndex which is the parent (i.e., physical) device of this virtual device on systems implementing the HOST-RESOURCES-MIB. The value zero denotes this virtual device is not any child represented in the hrStorageTable."

```
 ::= { vmStorageEntry 3 }
```

vmStorageSourceType OBJECT-TYPE

SYNTAX VirtualMachineStorageSourceType
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The source type of the virtual storage device."
::= { vmStorageEntry 4 }

vmStorageSourceTypeString OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..255))
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "A (detailed) textual string of the source type of the
 virtual storage device. For example, this represents
 the specific format name of the sparse file."
::= { vmStorageEntry 5 }

vmStorageResourceID OBJECT-TYPE

SYNTAX SnmpAdminString (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."
::= { vmStorageEntry 6 }

vmStorageAccess OBJECT-TYPE

SYNTAX VirtualMachineStorageAccess
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The access permission of the virtual storage device."
::= { vmStorageEntry 7 }

vmStorageMediaType OBJECT-TYPE

SYNTAX VirtualMachineStorageMediaType
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The media type of the virtual storage device."
::= { vmStorageEntry 8 }

vmStorageMediaTypeString OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..255))
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"A (detailed) textual string of the virtual storage media. For example, this represents the specific driver name of the emulated media such as `IDE' and `SCSI'."

::= { vmStorageEntry 9 }

vmStorageSizeUnit OBJECT-TYPE

SYNTAX Integer32 (1..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The multiplication unit for vmStorageDefinedSize and vmStorageAllocatedSize. For example, when this value is 1048576, the storage size unit for vmStorageDefinedSize and vmStorageAllocatedSize is MiB."

::= { vmStorageEntry 10 }

vmStorageDefinedSize OBJECT-TYPE

SYNTAX Integer32 (-1|0..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The defined virtual storage size defined in the unit designated by vmStorageSizeUnit. If this information is not available, this value shall be -1."

::= { vmStorageEntry 11 }

vmStorageAllocatedSize OBJECT-TYPE

SYNTAX Integer32 (-1|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 vmStorageSizeUnit. When the virtual storage is block device or raw file, this value and vmStorageDefinedSize are supposed to equal. This value must not be different from vmStorageDefinedSize when vmStorageSourceType is `block' or `raw'. If this information is not available, this value shall be -1."

::= { vmStorageEntry 12 }

vmStorageReadIOs OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of read I/O requests."


```
::= { vmStorageEntry 13 }
```

```
vmStorageWriteIOs OBJECT-TYPE
```

```
    SYNTAX          Counter64
```

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

```
    STATUS          current
```

```
    DESCRIPTION
```

```
        "The number of write I/O requests."
```

```
::= { vmStorageEntry 14 }
```

```
-- The virtual network interfaces on each virtual machine.
```

```
vmNetworkTable OBJECT-TYPE
```

```
    SYNTAX          SEQUENCE OF VmNetworkEntry
```

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

```
    STATUS          current
```

```
    DESCRIPTION
```

```
        "The conceptual table of virtual network interfaces  
        attached to the virtual machine."
```

```
::= { vmObjects 8 }
```

```
vmNetworkEntry OBJECT-TYPE
```

```
    SYNTAX          VmNetworkEntry
```

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

```
    STATUS          current
```

```
    DESCRIPTION
```

```
        "An entry for one virtual storage device attached to the  
        virtual machine."
```

```
    INDEX { vmIndex, vmNetworkIndex }
```

```
::= { vmNetworkTable 1 }
```

```
VmNetworkEntry ::=
```

```
    SEQUENCE {
```

```
        vmNetworkIndex          VirtualMachineNetworkIndex,
```

```
        vmNetworkIfIndex        InterfaceIndexOrZero,
```

```
        vmNetworkParent         InterfaceIndexOrZero,
```

```
        vmNetworkModel          SnmpAdminString,
```

```
        vmNetworkPhysAddress    PhysAddress
```

```
    }
```

```
vmNetworkIndex OBJECT-TYPE
```

```
    SYNTAX          VirtualMachineNetworkIndex
```

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

```
    STATUS          current
```

```
    DESCRIPTION
```

```
        "A unique value identifying a virtual network interface  
        allocated to the virtual machine."
```

```
::= { vmNetworkEntry 1 }
```



```
vmNetworkIfIndex 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."
    ::= { vmNetworkEntry 2 }

vmNetworkParent OBJECT-TYPE
    SYNTAX      InterfaceIndexOrZero
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The value of ifIndex which corresponds to the parent
        (i.e., physical) device of this virtual device on.  The
        value zero denotes this virtual device is not any child
        represented in the ifTable."
    ::= { vmNetworkEntry 3 }

vmNetworkModel OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (0..255))
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "A textual string containing the (emulated) model of
        virtual network interface.  For example, this value is
        'virtio' when the emulation driver model is virtio."
    ::= { vmNetworkEntry 4 }

vmNetworkPhysAddress OBJECT-TYPE
    SYNTAX      PhysAddress
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The MAC address of the virtual network interface."
    ::= { vmNetworkEntry 5 }

-- Notification definitions:
vmStarted NOTIFICATION-TYPE
    OBJECTS      {
        vmName,
        vmUUID,
        vmOperState
    }
    STATUS       current
```



```
DESCRIPTION
    "This notification is generated when a virtual machine
    has been started and the start process has reached a
    stable state (e.g., running or crashed)."
```

```
 ::= { vmNotifications 1 }
```

```
vmStopped NOTIFICATION-TYPE
    OBJECTS      {
        vmName,
        vmUUID,
        vmOperState
    }
    STATUS      current
    DESCRIPTION
        "This notification is generated when a virtual machine
        has been stopped and the shutdown process has reached a
        stable state (e.g., shutdown, shutoff or crashed)."
```

```
 ::= { vmNotifications 2 }
```

```
vmSuspended NOTIFICATION-TYPE
    OBJECTS      {
        vmName,
        vmUUID,
        vmOperState
    }
    STATUS      current
    DESCRIPTION
        "This notification is generated when a virtual machine
        has been suspended and the suspension process has
        reached a stable state (e.g., paused or crashed)."
```

```
 ::= { vmNotifications 3 }
```

```
vmResumed NOTIFICATION-TYPE
    OBJECTS      {
        vmName,
        vmUUID,
        vmOperState
    }
    STATUS      current
    DESCRIPTION
        "This notification is generated when a virtual machine
        has been resumed and the resumption process has reached
        a stable state (e.g., running or crashed)."
```

```
 ::= { vmNotifications 4 }
```

```
-- Compliance definitions:
vmGroups      OBJECT IDENTIFIER ::= { vmConformance 1 }
```



```
vmCompliances OBJECT IDENTIFIER ::= { vmConformance 2 }
```

```
vmFullCompliances MODULE-COMPLIANCE
```

```
    STATUS          current
```

```
    DESCRIPTION
```

```
        "Compliance statement for implementations supporting  
        read/write access, according to the object definitions."
```

```
    MODULE          -- this module
```

```
    MANDATORY-GROUPS {
```

```
        vmHypervisorGroup,  
        vmVirtualMachineGroup,  
        vmCpuGroup,  
        vmCpuAffinityGroup,  
        vmStorageGroup,  
        vmNetworkGroup,  
        vmNotificationGroup
```

```
    }
```

```
    ::= { vmCompliances 1 }
```

```
vmReadOnlyCompliances MODULE-COMPLIANCE
```

```
    STATUS          current
```

```
    DESCRIPTION
```

```
        "Compliance statement for implementations supporting  
        only readonly access."
```

```
    MODULE          -- this module
```

```
    MANDATORY-GROUPS {
```

```
        vmHypervisorGroup,  
        vmVirtualMachineGroup,  
        vmCpuGroup,  
        vmCpuAffinityGroup,  
        vmStorageGroup,  
        vmNetworkGroup,  
        vmNotificationGroup
```

```
    }
```

```
OBJECT vmAdminState
```

```
MIN-ACCESS read-only
```

```
DESCRIPTION
```

```
    "Write access is not required."
```

```
OBJECT vmAutoStart
```

```
MIN-ACCESS read-only
```

```
DESCRIPTION
```

```
    "Write access is not required."
```

```
OBJECT vmMinCpuNumber
```

```
MIN-ACCESS read-only
```

```
DESCRIPTION
```


"Write access is not required."

OBJECT vmMaxCpuNumber
MIN-ACCESS read-only
DESCRIPTION

"Write access is not required."

OBJECT vmMinMem
MIN-ACCESS read-only
DESCRIPTION

"Write access is not required."

OBJECT vmMaxMem
MIN-ACCESS read-only
DESCRIPTION

"Write access is not required."

OBJECT vmCpuAffinity
MIN-ACCESS read-only
DESCRIPTION

"Write access is not required."

::= { vmCompliances 2 }

vmHypervisorGroup OBJECT-GROUP

OBJECTS {
 vmHvSoftware,
 vmHvVersion,
 vmHvObjectID,
 vmHvUpTime,
 vmNumber,
 vmTableLastChange
}

STATUS current

DESCRIPTION

"A collection of objects providing insight into the
hypervisor itself."

::= { vmGroups 1 }

vmVirtualMachineGroup OBJECT-GROUP

OBJECTS {
 -- vmIndex
 vmName,
 vmUUID,
 vmOSType,
 vmAdminState,
 vmOperState,
 vmAutoStart,
 vmPersistent,


```
        vmCurCpuNumber,
        vmMinCpuNumber,
        vmMaxCpuNumber,
        vmMemUnit,
        vmCurMem,
        vmMinMem,
        vmMaxMem,
        vmUpTime,
        vmCpuTime
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing insight into the
        virtual machines) controlled by a hypervisor."
    ::= { vmGroups 2 }

vmCpuGroup OBJECT-GROUP
    OBJECTS {
        -- vmCpuIndex,
        vmCpuCoreTime
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing insight into the
        virtual machines) controlled by a hypervisor."
    ::= { vmGroups 3 }

vmCpuAffinityGroup OBJECT-GROUP
    OBJECTS {
        -- vmCpuPhysIndex,
        vmCpuAffinity
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing insight into the
        virtual machines) controlled by a hypervisor."
    ::= { vmGroups 4 }

vmStorageGroup OBJECT-GROUP
    OBJECTS {
        -- vmStorageVmIndex,
        -- vmStorageIndex,
        vmStorageParent,
        vmStorageSourceType,
        vmStorageSourceTypeString,
        vmStorageResourceID,
        vmStorageAccess,
        vmStorageMediaType,
```



```
        vmStorageMediaTypeString,
        vmStorageSizeUnit,
        vmStorageDefinedSize,
        vmStorageAllocatedSize,
        vmStorageReadIOs,
        vmStorageWriteIOs
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing insight into the
        virtual storage devices controlled by a hypervisor."
    ::= { vmGroups 5 }

vmNetworkGroup OBJECT-GROUP
    OBJECTS {
        -- vmNetworkIndex,
        vmNetworkIfIndex,
        vmNetworkParent,
        vmNetworkModel,
        vmNetworkPhysAddress
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing insight into the
        virtual network interfaces controlled by a hypervisor."
    ::= { vmGroups 6 }

vmNotificationGroup NOTIFICATION-GROUP
    NOTIFICATIONS {
        vmStarted,
        vmStopped,
        vmSuspended,
        vmResumed
    }
    STATUS          current
    DESCRIPTION
        "A collection of notifications for virtual machines
        controlled by a hypervisor."
    ::= { vmGroups 7 }

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 -----
vm-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 vmHvSoftware and vmHvVersion 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, vmCpuTable, vmCpuAffinityTable, vmStorageTable and vmNetworkTable list information about the virtual machines and their virtual resource allocation. 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, vmMinCpuNumber, vmMaxCpuNumber, vmMinMem, vmMaxMem, and vmCpuAffinity 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 [[RFC3414](#)] and the View-based Access Control Model [[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. References

6.1. Normative References

- [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.
- [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.
- [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.

6.2. Informative References

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

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

Michael MacFaden
VMware Inc.

Email: mrm@vmware.com

Juergen Schoenwaelder
Jacobs University
Campus Ring 1
Bremen 28759
Germany

Email: j.schoenwaelder@jacobs-university.de

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

Tina Tsou
Huawei Technologies (USA)
2330 Central Expressway
Santa Clara CA 95050
USA

Email: tina.tsou.zouting@huawei.com

Cathy Zhou
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R. China

Email: cathyzhou@huawei.com

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

