



## 業界標準の管理情報ベース

この章では、Cisco Unified Communications Manager (Cisco Unified CM) でサポートされ、Simple Network Management Protocol (SNMP; 簡易ネットワーク管理プロトコル) で使用される業界標準の Management Information Base (MIB; 管理情報ベース) テキスト ファイルについて説明します。内容は、次のとおりです。

- 「SYSAPPL-MIB」 (P.8-1)
- 「RFC1213-MIB (MIB-II)」 (P.8-28)
- 「HOST-RESOURCES-MIB」 (P.8-73)
- 「IF-MIB」 (P.8-106)

## SYSAPPL-MIB



(注)

この SYSAPPL-MIB は形式が変更されています。この項のすべての MIB は、<http://tools.cisco.com/Support/SNMP/do/BrowseMIB.do?local=en&step=2> からダウンロードしてコンパイルしてください。

MIB モジュールは、ホスト システムにインストールされて実行される実行可能ファイルおよびファイルのコレクションとしてアプリケーションをモデル化する管理オブジェクトを定義します。MIB は、アプリケーションのシステムレベルのビューを表します。つまり、この MIB 内のオブジェクトは、アプリケーションに特殊な機器を追加しなくてもシステム自体から通常取得可能な属性に限定されます。

SYSAPPL-MIB をコンパイルする前に、次のリストの MIB を、リストされている順にコンパイルする必要があります。

1. RFC1155-SMI
2. RFC-1212
3. SNMPv2-SMI-v1
4. SNMPv2-TC-v1
5. SYSAPPL-MIB

さらに、次のファイルをダウンロードします。

- OID ファイル : SYSAPPL-MIB.oid

この項の内容は次のとおりです。

- 「改訂」 (P.8-2)
- 「定義」 (P.8-2)
- 「システム アプリケーション MIB」 (P.8-2)
- 「テキストの表記法」 (P.8-3)
- 「インストールされたアプリケーション グループ」 (P.8-3)
- 「テーブル サイズを制御する追加のスカラ オブジェクト」 (P.8-21)
- 「適合マクロ」 (P.8-25)
- 「トラブルシューティング」 (P.8-26)

## 改訂

表 8-1 に、最新の改訂から順に MIB の改訂を示します。

表 8-1 改訂の履歴

日付	処置	説明
1997 年 10 月 20 日	IETF アプリケーション MIB ワーキング グループ	::= { mib-2 54 }

## 定義

次の定義が SYSAPP-MIB 用にインポートされています。

- MODULE-IDENTITY、OBJECT-TYPE、mib-2、Unsigned32 (SNMPv2-SMI が入手可能になるまで当分の間 CISCO-TC から取得)、Unsigned32、TimeTicks、Counter32、Gauge32 TimeTicks、Counter32、Gauge32
- SNMPv2-SMI から : Unsigned32
- CISCO-TC から : DateAndTime、TEXTUAL-CONVENTION
- SNMPv2-TC から : MODULE-COMPLIANCE、OBJECT-GROUP
- SNMPv2-CONF から

## システム アプリケーション MIB

sysApplMIB MODULE-IDENTITY

sysApplOBJ OBJECT IDENTIFIER ::= { sysApplMIB 1 }

sysApplInstalled OBJECT IDENTIFIER ::= { sysApplOBJ 1 }

sysApplRun OBJECT IDENTIFIER ::= { sysApplOBJ 2 }

sysApplMap OBJECT IDENTIFIER ::= { sysApplOBJ 3 }

sysApplNotifications OBJECT IDENTIFIER ::= { sysApplMIB 2 }

sysApplConformance OBJECT IDENTIFIER ::= { sysApplMIB 3 }

## テキストの表記法

### RunState ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

This TC describes the current execution state of a running application or process. The possible values are: running(1), runnable(2), waiting for a resource (CPU, etc.) waiting(3), waiting for an event exiting(4), other(5) other invalid state.

SYNTAX INTEGER { running (1); runnable (2); waiting for resource and waiting (3); waiting for event and exiting (4); other (5) }

### LongUtf8String ::= TEXTUAL-CONVENTION

DISPLAY-HINT 1024a

STATUS current

DESCRIPTION

To facilitate internationalization, this TC represents information taken from the ISO/IEC IS 10646-1 character set, encoded as an octet string using the UTF-8 character encoding scheme described in RFC 2044 [10]. For strings in 7-bit US-ASCII, there is no impact since the UTF-8 representation is identical to the US-ASCII encoding.

SYNTAX OCTET STRING (SIZE (0..1024))

### Utf8String ::= TEXTUAL-CONVENTION

DISPLAY-HINT 255a

STATUS current

DESCRIPTION

To facilitate internationalization, this TC represents information taken from the ISO/IEC IS 10646-1 character set, encoded as an octet string using the UTF-8 character encoding scheme described in RFC 2044 [10]. For strings in 7-bit US-ASCII, there is no impact since the UTF-8 representation is identical to the US-ASCII encoding.

SYNTAX OCTET STRING (SIZE (0..255))

## インストールされたアプリケーション グループ

このグループは、ホスト コンピュータにインストールされたアプリケーション パッケージについての情報を提供します。グループには、次のような 2 つのテーブルが含まれています。

- `sysApplInstallPkgTable` : アプリケーション パッケージを記述します。
- `sysApplInstallElmtTable` : アプリケーション パッケージを構成する構成要素 (ファイルおよび実行可能ファイル) を記述します。

グループに表示されるためには、アプリケーションとそのコンポーネント ファイルがシステム自体によって (場合によっては、一種のソフトウェア インストール メカニズムまたはレジストリを使って) 検出可能でなければなりません。

## sysApplInstallPkgTable

システムによりインストールされたアプリケーションパッケージテーブルは、システムにインストールされたソフトウェアパッケージの情報を提供します。これらのパッケージは、実行可能ファイルやそれ以外のファイルなど、さまざまなファイルで構成されることがあります。

### sysApplInstallPkgTable OBJECT-TYPE

SYNTAX SysApplInstallPkgEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table listing the software application packages installed on a host computer. In order to appear in this table, it may be necessary for the application to be installed using some type of software installation mechanism or global registry so that its existence can be detected by the agent implementation.

::= { sysApplInstalled 1 }

### sysApplInstallPkgEntry OBJECT-TYPE

SYNTAX SysApplInstallPkgEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The logical row describing an installed application package.

INDEX { sysApplInstallPkgIndex }

::= { sysApplInstallPkgTable 1 }

SysApplInstallPkgEntry ::= SEQUENCE { sysApplInstallPkgIndex Unsigned32, sysApplInstallPkgManufacturer Utf8String, sysApplInstallPkgProductName Utf8String, sysApplInstallPkgVersion Utf8String, sysApplInstallPkgSerialNumber Utf8String, sysApplInstallPkgDate DateAndTime, sysApplInstallPkgLocation LongUtf8String }

### sysApplInstallPkgIndex OBJECT-TYPE

SYNTAX Unsigned32 (1..ffffffffh)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An integer used only for indexing purposes. Generally monotonically increasing from 1 as new applications are installed. The value for each installed application must remain constant at least from one re-initialization of the network management entity which implements this MIB module to the next re-initialization. The specific value is meaningful only within a given SNMP entity. A sysApplInstallPkgIndex value must not be re-used until the next agent entity restart in the event the installed application entry is deleted.

::= { sysApplInstallPkgEntry 1 }

### sysApplInstallPkgManufacturer OBJECT-TYPE

SYNTAX Utf8String

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The Manufacturer of the software application package.

::= { sysAppInstallPkgEntry 2 }

**sysAppInstallPkgProductName OBJECT-TYPE**

SYNTAX Utf8String

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The name assigned to the software application package by the Manufacturer.

::= { sysAppInstallPkgEntry 3 }

**sysAppInstallPkgVersion OBJECT-TYPE**

SYNTAX Utf8String

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The version number assigned to the application package by the manufacturer of the software.

::= { sysAppInstallPkgEntry 4 }

**sysAppInstallPkgSerialNumber OBJECT-TYPE**

SYNTAX Utf8String

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The serial number of the software assigned by the manufacturer.

::= { sysAppInstallPkgEntry 5 }

**sysAppInstallPkgDate OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The date and time this software application was installed on the host.

::= { sysAppInstallPkgEntry 6 }

**sysAppInstallPkgLocation OBJECT-TYPE**

SYNTAX LongUtf8String

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The complete path name where the application package is installed. For example, the value would be /opt/MyapplDir if the application package was installed in the /opt/MyapplDir directory.

```
::= { sysAppInstallPkgEntry 7 }
```

## sysAppInstallElmtTable

このテーブルは、sysAppInstallPkg テーブルで定義されたアプリケーションを構成するホスト コンピュータにインストールされた個々のアプリケーション パッケージ要素（ファイルおよび実行可能ファイル）を詳述します。このテーブル内の各エントリには、一部となっているアプリケーション パッケージを識別するために、sysAppInstallPkg テーブルへのインデックスがあります。このため、sysAppInstallPkg テーブルのインスタンスごとに多数のエントリがこのテーブルに存在する場合があります。

テーブル エントリは、インストールされた特定のアプリケーション パッケージに関連付けられたすべての要素を取得しやすくするため、sysAppInstallPkgIndex、sysAppInstallElmtIndex によりインデックスが付けられます。

### sysAppInstallElmtTable OBJECT-TYPE

SYNTAX SEQUENCE OF SysAppInstallElmtEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

This table details the individual application package elements (files and executables) installed on the host computer which comprise the applications defined in the sysAppInstallPkg Table. Each entry in this table has an index to the sysAppInstallPkg table to identify the application package of which it is a part. As a result, there may be many entries in this table for each instance in the sysAppInstallPkg Table.

Table entries are indexed by sysAppInstallPkgIndex, sysAppInstallElmtIndex to facilitate retrieval of all elements associated with a particular installed application package.

```
::= { sysAppInstalled 2 }
```

### sysAppInstallElmtEntry OBJECT-TYPE

SYNTAX SysAppInstallElmtEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The logical row describing an element of an installed application. The element may be an executable or non-executable file.

INDEX {sysAppInstallPkgIndex, sysAppInstallElmtIndex}

```
::= { sysAppInstallElmtTable 1 }
```

```
SysAppInstallElmtEntry ::= SEQUENCE {
    sysAppInstallElmtIndex Unsigned32,
    sysAppInstallElmtNameUtf8String,
    sysAppInstallElmtTypeINTEGER,
    sysAppInstallElmtDateDateAndTime,
    sysAppInstallElmtPathLongUtf8String,
    sysAppInstallElmtSizeHighUnsigned32,
    sysAppInstallElmtSizeLow Unsigned32,
    sysAppInstallElmtRoleBITS,
    sysAppInstallElmtRoleOCTET STRING,
    sysAppInstallElmtModifyDate DateAndTime,
    sysAppInstallElmtCurSizeHighUnsigned32,
    sysAppInstallElmtCurSizeLow Unsigned32 }
```

**sysApplInstallElmtIndex OBJECT-TYPE**

SYNTAX Unsigned32 (1...fffffffh)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An arbitrary integer used for indexing. The value of this index is unique among all rows in this table that exist or have existed since the last agent restart.

::= { sysApplInstallElmtEntry 1 }

**sysApplInstallElmtName OBJECT-TYPE**

SYNTAX Utf8String

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The name of this element which is contained in the application.

::= { sysApplInstallElmtEntry 2 }

**sysApplInstallElmtType OBJECT-TYPE**

SYNTAX INTEGER { unknown(1), nonexecutable(2), operatingSystem(3), executable deviceDriver(4), executable application(5), executable }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The type of element that is part of the installed application.

::= { sysApplInstallElmtEntry 3 }

**sysApplInstallElmtDate OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The date and time that this component was installed on the system.

::= { sysApplInstallElmtEntry 4 }

**sysApplInstallElmtPath OBJECT-TYPE**

SYNTAX LongUtf8String

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The full directory path where this element is installed. For example, the value would be /opt/EMPuma/bin for an element installed in the directory /opt/EMPuma/bin. Most application packages include information about the elements contained in the package. In addition, elements are typically installed in sub-directories under the package installation directory. In cases where the

element path names are not included in the package information itself, the path can usually be determined by a simple search of the sub-directories. If the element is not installed in that location and there is no other information available to the agent implementation, then the path is unknown and null is returned.

::= { sysAppInstallElmtEntry 5 }

#### **sysAppInstallElmtSizeHigh OBJECT-TYPE**

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The installed file size in  $2^{32}$  byte blocks. This is the size of the file on disk immediately after installation. For example, for a file with a total size of 4,294,967,296 bytes, this variable would have a value of 1; for a file with a total size of 4,294,967,295 bytes this variable would be 0.

::= { sysAppInstallElmtEntry 6 }

#### **sysAppInstallElmtSizeLow OBJECT-TYPE**

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The installed file size modulo  $2^{32}$  bytes. This is the size of the file on disk immediately after installation. For example, for a file with a total size of 4,294,967,296 bytes this variable would have a value of 0; for a file with a total size of 4,294,967,295 bytes this variable would be 4,294,967,295.

::= { sysAppInstallElmtEntry 7 }

#### **sysAppInstallElmtRole OBJECT-TYPE**

SYNTAX OCTET STRING (SIZE(1))

SYNTAX BITS { executable (0), exclusive (1), primary (2), required (3), dependent (4), unknown(5) }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

An operator assigned value used in the determination of application status. This value is used by the agent to determine both the mapping of started processes to the initiation of an application, as well as to allow for a determination of application health. The default value, unknown(5), is used when an operator has not yet assigned one of the other values. If unknown(5) is set, bits 1 - 4 have no meaning. The possible values are:

- executable (0)—An application may have one or more executable elements. The rest of the bits have no meaning if the element is not executable.
- exclusive(1)—Only one copy of an exclusive element may be running per invocation of the running application.



- primary(2)—The primary executable. An application can have one, and only one element that is designated as the primary executable. The execution of this element constitutes an invocation of the application. This is used by the agent implementation to determine the initiation of an application. The primary executable must remain running long enough for the agent implementation to detect its presence.
- required(3)—An application may have zero or more required elements. All required elements must be running in order for the application to be judged to be running and healthy.
- dependent(4)—An application may have zero or more dependent elements. Dependent elements may not be running unless required elements are.
- unknown(5)—Default value for the case when an operator has not yet assigned one of the other values. When set, bits 1, 2, 3, and 4 have no meaning.

sysAppInstallElmtRole is used by the agent implementation in determining the initiation of an application, the current state of a running application (see sysAppRunCurrentState), when an application invocation is no longer running, and the exit status of a terminated application invocation (see sysAppPastRunExitState).

--DEFVAL { 5 }

::= { sysAppInstallElmtEntry 8 }

#### **sysAppInstallElmtModifyDate OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The date and time that this element was last modified. Modification of the sysAppInstallElmtRole columnar object does NOT constitute a modification of the element itself and should not affect the value of this object.

::= { sysAppInstallElmtEntry 9 }

#### **sysAppInstallElmtCurSizeHigh OBJECT-TYPE**

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The current file size in  $2^{32}$  byte blocks. For example, for a file with a total size of 4,294,967,296 bytes, this variable would have a value of 1; for a file with a total size of 4,294,967,295 bytes this variable would be 0.

::= { sysAppInstallElmtEntry 10 }

#### **sysAppInstallElmtCurSizeLow OBJECT-TYPE**

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The current file size modulo  $2^{32}$  bytes. For example, for a file with a total size of 4,294,967,296 bytes this variable would have a value of 0; for a file with a total size of 4,294,967,295 bytes this variable would be 4,294,967,295.

```
::= { sysAppInstallElmtEntry 11 }
```

## sysApplRun グループ

このグループは、ホストシステムで現在実行中か以前に実行された、呼び出されたアプリケーションのアクティビティ情報をモデル化します。同様に、現在実行中のプロセスと、過去に実行されたプロセスを示すため、呼び出されたアプリケーションの個々の要素もモデル化されます。

## sysApplRunTable

sysApplRunTable には、ホストで現在実行中のアプリケーション インスタンスが格納されます。1 つのアプリケーションが複数回呼び出されることがあるため、アプリケーションの INVOCATION ごとにこのテーブルにエントリが追加されます。テーブルは、マネージャが特定のアプリケーション パッケージのすべての呼び出しを簡単に特定できるように、sysAppInstallPkgIndex、sysApplRunIndex によりインデックスが付けられます。

### sysApplRunTable OBJECT-TYPE

SYNTAX SEQUENCE OF SysApplRunEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table describes the applications which are executing on the host. Each time an application is invoked, an entry is created in this table. When an application ends, the entry is removed from this table and a corresponding entry is created in the SysAppIPastRunTable.

A new entry is created in this table whenever the agent implementation detects a new running process that is an installed application element whose sysAppInstallElmtRole designates it as being the application's primary executable (sysAppInstallElmtRole = primary(2)).

The table is indexed by sysAppInstallPkgIndex, sysApplRunIndex to enable managers to easily locate all invocations of a particular application package.

```
::= { sysApplRun 1 }
```

### sysApplRunEntry OBJECT-TYPE

SYNTAX SysApplRunEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The logical row describing an application which is currently running on this host.

INDEX { sysAppInstallPkgIndex, sysApplRunIndex }

```
::= { sysApplRunTable 1 }
```

```
SysApplRunEntry ::= SEQUENCE { sysApplRunIndex Unsigned32, sysApplRunStarted DateAndTime, sysApplRunCurrentState RunState }
```

### sysApplRunIndex OBJECT-TYPE

SYNTAX Unsigned32 (1..'ffffff'h)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

Part of the index for this table. An arbitrary integer used only for indexing purposes. Generally monotonically increasing from 1 as new applications are started on the host, it uniquely identifies application invocations.

The numbering for this index increases by 1 for each INVOCATION of an application, regardless of which installed application package this entry represents a running instance of. An example of the indexing for a couple of entries is shown below.

sysApplRunStarted.17.14

sysApplRunStarted.17.63

sysApplRunStarted.18.13

:

In this example, the agent has observed 12 application invocations when the application represented by entry 18 in the sysApplInstallPkgTable is invoked. The next invocation detected by the agent is an invocation of installed application package 17. Some time later, installed application 17 is invoked a second time.



(注) このインデックスは、アプリケーション呼び出しのリアルタイム（壁掛け時計）の順序付けを反映するためのものではありません。アプリケーションの実行中のインスタンスを一意に識別するためだけです。sysApplInstallPkgIndex はこのテーブルの INDEX 句に含まれていますが、インストールされたアプリケーションによるこのテーブルの検索を簡単にするためだけに機能し、テーブル エントリを一意に識別するために利用することはできません。

::= { sysApplRunEntry 1 }

#### sysApplRunStarted OBJECT-TYPE

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The date and time that the application was started.

::= { sysApplRunEntry 2 }

#### sysApplRunCurrentState OBJECT-TYPE

SYNTAX RunState

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The current state of the running application instance. The possible values are running(1), runnable(2) but waiting for a resource such as CPU, waiting(3) for an event, exiting(4), or other(5). This value is based on an evaluation of the running elements of this application instance (see sysApplElmRunState) and their Roles as defined by sysApplInstallElmtRole. An agent

implementation may detect that an application instance is in the process of exiting if one or more of its REQUIRED elements are no longer running. Most agent implementations will wait until a second internal poll has been completed to give the system time to start REQUIRED elements before marking the application instance as exiting.

```
::= { sysApplRunEntry 3 }
```

## sysApplPastRunTable

sysApplPastRunTable は、ホスト コンピュータで以前に実行されたアプリケーションの履歴を提供します。エント리는 sysApplRunTable から削除され、アプリケーションが非アクティブになると対応するエント리가このテーブルに追加されます。テーブル サイズが sysApplPastRunMaxRows により決定された最大に達するか、sysApplPastRunTblTimeLimit により設定された時間制限をエント리가超過すると、エント리는期限切れになるまでこのテーブルに残されます。

エント리의期限が切れると、sysApplPastRunTimeEnded の値により判断された最も古いエント리가最初に削除されます。

### sysApplPastRunTable OBJECT-TYPE

SYNTAX SEQUENCE OF SysApplPastRunEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

A history of the applications that have previously run on the host computer. An entry's information is moved to this table from the sysApplRunTable when the invoked application represented by the entry ceases to be running. An agent implementation can determine that an application invocation is no longer running by evaluating the running elements of the application instance and their Roles as defined by sysApplInstallElmtRole. Obviously, if there are no running elements for the application instance, then the application invocation is no longer running.

If any one of the REQUIRED elements is not running, the application instance may be in the process of exiting. Most agent implementations will wait until a second internal poll has been completed to give the system time to either restart partial failures or to give all elements time to exit. If, after the second poll, there are REQUIRED elements that are not running, then the application instance may be considered by the agent implementation to no longer be running.

Entries remain in the sysApplPastRunTable until they are aged out when either the table size reaches a maximum as determined by the sysApplPastRunMaxRows, or when an entry has aged to exceed a time limit as set by sysApplPastRunTblTimeLimit.

Entries in this table are indexed by sysApplInstallPkgIndex, sysApplPastRunIndex to facilitate retrieval of all past run invocations of a particular installed application.

```
::= { sysApplRun 2 }
```

### sysApplPastRunEntry OBJECT-TYPE

SYNTAX SysApplPastRunEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The logical row describing an invocation of an application which was previously run and has terminated. The entry is basically copied from the sysApplRunTable when the application instance terminates. Hence, the entry's value for sysApplPastRunIndex is the same as its value was for sysApplRunIndex.

INDEX { sysApplInstallPkgIndex, sysApplPastRunIndex }

::= { sysApplPastRunTable 1 }

SysApplPastRunEntry ::= SEQUENCE { sysApplPastRunIndex Unsigned32,  
sysApplPastRunStarted DateAndTime, sysApplPastRunExitState INTEGER,  
sysApplPastRunTimeEnded DateAndTime

#### sysApplPastRunIndex OBJECT-TYPE

SYNTAX Unsigned32 (1...ffffffh)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

Part of the index for this table. An integer matching the value of the removed sysApplRunIndex corresponding to this row.

::= { sysApplPastRunEntry 1 }

#### sysApplPastRunStarted OBJECT-TYPE

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The date and time that the application was started.

::= { sysApplPastRunEntry 2 }

#### sysApplPastRunExitState OBJECT-TYPE

SYNTAX INTEGER { complete (1), failed (2), other (3) }

- complete (1)—normal exit at sysApplRunTimeEnded
- failed (2)—abnormal exit
- other (3)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The state of the application instance when it terminated. This value is based on an evaluation of the running elements of an application and their Roles as defined by sysApplInstallElmtRole. An application instance is said to have exited in a COMPLETE state and its entry is removed from the sysApplRunTable and added to the sysApplPastRunTable when the agent detects that ALL elements of an application invocation are no longer running. Most agent implementations will wait until a second internal poll has been completed to give the system time to either restart partial failures or to give all elements time to exit. A failed state occurs if, after the second poll, any elements continue to run but one or more of the REQUIRED elements are no longer running.

All other combinations MUST be defined as OTHER.

```
::= { sysApplPastRunEntry 3 }
```

#### **sysApplPastRunTimeEnded OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The DateAndTime the application instance was determined to be no longer running.

```
::= { sysApplPastRunEntry 4 }
```

## **sysApplElmtRunTable**

sysApplElmtRunTable には、ホストで現在実行中の各プロセスのエントリが格納されています。エントリは、各プロセスの起動時にプロセスごとにこのテーブルに作成され、プロセスが終了するまでテーブルに残されます。テーブルは、システムにインストールされた、特定の呼び出されたアプリケーションの実行中の要素をすべて特定しやすくするために、sysApplElmtRunInstallPkg、sysApplElmtRunInvocID、および sysApplElmtRunIndex によりインデックスが付けられます。

#### **sysApplElmtRunTable OBJECT-TYPE**

SYNTAX SEQUENCE OF SysApplElmtRunEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table describes the processes which are currently executing on the host system. Each entry represents a running process and is associated with the invoked application of which that process is a part, if possible. This table contains an entry for every process currently running on the system, regardless of whether its 'parent' application can be determined. So, for example, processes like 'ps' and 'grep' will have entries though they are not associated with an installed application package.

Because a running application may involve more than one executable, it is possible to have multiple entries in this table for each application. Entries are removed from this table when the process terminates. The table is indexed by sysApplElmtRunInstallPkg, sysApplElmtRunInvocID, and sysApplElmtRunIndex to facilitate the retrieval of all running elements of a particular invoked application which has been installed on the system.

```
::= { sysApplRun 3 }
```

#### **sysApplElmtRunEntry OBJECT-TYPE**

SYNTAX SysApplElmtRunEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The logical row describing a process currently running on this host. When possible, the entry is associated with the invoked application of which it is a part.

INDEX { sysApplElmtRunInstallPkg, sysApplElmtRunInvocID, sysApplElmtRunIndex }

```
::= { sysApplElmtRunTable 1 }
```

```
SysAppElmtRunEntry ::= SEQUENCE {sysAppElmtRunInstallPkg Unsigned32,
sysAppElmtRunInvocIDUnsigned32, sysAppElmtRunIndex Unsigned32,
sysAppElmtRunInstallID Unsigned32, sysAppElmtRunTimeStartedDateAndTime,
sysAppElmtRunState RunState, sysAppElmtRunNameLongUtf8String,
sysAppElmtRunParameters Utf8String, sysAppElmtRunCPU TimeTicks,
sysAppElmtRunMemory Gauge32, sysAppElmtRunNumFiles Gauge32,
sysAppElmtRunUserUtf8String }
```

**sysAppElmtRunInstallPkg OBJECT-TYPE**

SYNTAX Unsigned32 (0...ffffffh)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

Part of the index for this table, this value identifies the installed software package for the application of which this process is a part. Provided that the process's 'parent' application can be determined, the value of this object is the same value as the sysAppInstallPkgIndex for the entry in the sysAppInstallPkgTable that corresponds to the installed application of which this process is a part.

If, however, the 'parent' application cannot be determined, (for example the process is not part of a particular installed application), the value for this object is then '0', signifying that this process cannot be related back to an application, and in turn, an installed software package.

::= { sysAppElmtRunEntry 1 }

**sysAppElmtRunInvocID OBJECT-TYPE**

SYNTAX Unsigned32 (0...ffffffh)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

Part of the index for this table, this value identifies the invocation of an application of which this process is a part. Provided that the 'parent' application can be determined, the value of this object is the same value as the sysAppRunIndex for the corresponding application invocation in the sysAppRunTable.

If, however, the 'parent' application cannot be determined, the value for this object is then '0', signifying that this process cannot be related back to an invocation of an application in the sysAppRunTable.

::= { sysAppElmtRunEntry 2 }

**sysAppElmtRunIndex OBJECT-TYPE**

SYNTAX Unsigned32 (0...ffffffh)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

Part of the index for this table. A unique value for each process running on the host. Wherever possible, this should be the system's native, unique identification number.

::= { sysAppElmtRunEntry 3 }

**sysAppElmtRunInstallID OBJECT-TYPE**

SYNTAX Unsigned32 (0...ffffffh)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The index into the sysAppInstallElmtTable. The value of this object is the same value as the sysAppInstallElmtIndex for the application element of which this entry represents a running instance.

If this process cannot be associated with an installed executable, the value should be '0'.

::= { sysAppElmtRunEntry 4 }

#### **sysAppElmtRunTimeStarted OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The time the process was started.

::= { sysAppElmtRunEntry 5 }

#### **sysAppElmtRunState OBJECT-TYPE**

SYNTAX RunState

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The current state of the running process. The possible values are running(1), runnable(2) but waiting for a resource such as CPU, waiting(3) for an event, exiting(4), or other(5).

::= { sysAppElmtRunEntry 6 }

#### **sysAppElmtRunName OBJECT-TYPE**

SYNTAX LongUtf8String

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The full path and filename of the process. For example, /opt/MYYpkg/bin/myyproc would be returned for process myyproc whose execution path is /opt/MYYpkg/bin/myyproc.

::= { sysAppElmtRunEntry 7 }

#### **sysAppElmtRunParameters OBJECT-TYPE**

SYNTAX Utf8String

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The starting parameters for the process.

::= { sysAppElmtRunEntry 8 }

#### **sysAppElmtRunCPU OBJECT-TYPE**



SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of centi-seconds of the total system CPU resources consumed by this process. Note that on a multi-processor system, this value may have been incremented by more than one centi-second in one centi-second of real (wall clock) time.

::= { sysAppElmtRunEntry 9 }

#### **sysAppElmtRunMemory OBJECT-TYPE**

SYNTAX Gauge32

UNITS Kbytes

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The total amount of real system memory measured in Kbytes currently allocated to this process.

::= { sysAppElmtRunEntry 10 }

#### **sysAppElmtRunNumFiles OBJECT-TYPE**

SYNTAX Gauge32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of regular files currently open by the process. Transport connections (sockets) should NOT be included in the calculation of this value, nor should operating system specific special file types.

::= { sysAppElmtRunEntry 11 }

#### **sysAppElmtRunUser OBJECT-TYPE**

SYNTAX Utf8String

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The process owner's login name (e.g. root).

::= { sysAppElmtRunEntry 12 }

## **sysAppElmtPastRunTable**

`sysAppElmtPastRunTable` は、アプリケーションの一部としてホストで以前に実行されたプロセスの履歴を保持します。プロセスの終了時に、プロセスを表すエントリが `sysAppElmtRunTable` から削除され、プロセスが識別可能なアプリケーションの一部であった場合は、対応するエントリがこのテーブルに作成されます。プロセスを呼び出されたアプリケーションに関連付けることができなかった場合、対応するエントリは作成されません。

したがって、sysAppElmtRunTable にシステムで現在実行中のすべてのプロセスのエントリが格納されている一方で、sysAppElmtPastRunTable には呼び出されたアプリケーションの一部として以前に実行されたプロセスのエントリだけが格納されています。

テーブル内のエントリの数が sysAppElmtPastRunMaxRows により決定された最大に達するか、sysAppElmtPastRunTblTimeLimit により設定された時間制限をエントリが超過すると、エントリは期限切れになるまでこのテーブルに残されます。

エントリの期限が切れると、sysAppElmtPastRunTimeEnded の値により判断された最も古いエントリが最初に削除されます。

テーブルは、システムにインストールされた、特定の呼び出されたアプリケーションの以前に実行されたプロセスをすべて特定しやすくするために、sysAppInstallPkgIndex (sysAppInstallPkgTable から)、sysAppElmtPastRunInvocID、および sysAppElmtPastRunIndex によりインデックスが付けられます。

#### sysAppElmtPastRunTable OBJECT-TYPE

SYNTAX SEQUENCE OF SysAppElmtPastRunEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table describes the processes which have previously executed on the host system as part of an application. Each entry represents a process which has previously executed and is associated with the invoked application of which it was a part. Because an invoked application may involve more than one executable, it is possible to have multiple entries in this table for each application invocation. Entries are added to this table when the corresponding process in the sysAppElmtRunTable terminates.

Entries remain in this table until they are aged out when either the number of entries in the table reaches a maximum as determined by sysAppElmtPastRunMaxRows, or when an entry has aged to exceed a time limit as set by sysAppElmtPastRunTblTimeLimit. When aging out entries, the oldest entry, as determined by the value of sysAppElmtPastRunTimeEnded, will be removed first.

The table is indexed by sysAppInstallPkgIndex (from the sysAppInstallPkgTable), sysAppElmtPastRunInvocID, and sysAppElmtPastRunIndex to make it easy to locate all previously executed processes of a particular invoked application that has been installed on the system.

::= { sysAppRun 4 }

#### sysAppElmtPastRunEntry OBJECT-TYPE

SYNTAX SysAppElmtPastRunEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The logical row describing a process which was previously executed on this host as part of an installed application. The entry is basically copied from the sysAppElmtRunTable when the process terminates. Hence, the entry's value for sysAppElmtPastRunIndex is the same as its value was for sysAppElmtRunIndex. Note carefully: only those processes which could be associated with an identified application are included in this table.

INDEX { sysAppInstallPkgIndex, sysAppElmtPastRunInvocID, sysAppElmtPastRunIndex }

::= { sysAppElmtPastRunTable 1 }

```
SysAppElmtPastRunEntry ::= SEQUENCE { sysAppElmtPastRunInvocIDUnsigned32,
sysAppElmtPastRunIndex Unsigned32, sysAppElmtPastRunInstallID Unsigned32,
sysAppElmtPastRunTimeStartedDateAndTime, sysAppElmtPastRunTimeEnded DateAndTime,
sysAppElmtPastRunNameLongUtf8String, sysAppElmtPastRunParameters Utf8String,
sysAppElmtPastRunCPU TimeTicks, sysAppElmtPastRunMemory Unsigned32,
sysAppElmtPastRunNumFiles Unsigned32, sysAppElmtPastRunUserUtf8String }
```

**sysAppElmtPastRunInvocID OBJECT-TYPE**

SYNTAX Unsigned32 (1..ffffffh)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

Part of the index for this table, this value identifies the invocation of an application of which the process represented by this entry was a part. The value of this object is the same value as the sysAppRunIndex for the corresponding application invocation in the sysAppRunTable. If the invoked application as a whole has terminated, it will be the same as the sysAppPastRunIndex.

::= { sysAppElmtPastRunEntry 1 }

**sysAppElmtPastRunIndex OBJECT-TYPE**

SYNTAX Unsigned32 (0..'ffffff'h)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

Part of the index for this table. An integer assigned by the agent equal to the corresponding sysAppElmtRunIndex which was removed from the sysAppElmtRunTable and moved to this table when the element terminated. Note that entries in this table are indexed by sysAppElmtPastRunInvocID, sysAppElmtPastRunIndex.

The possibility exists, though unlikely, of a collision occurring by a new entry which was run by the same invoked application (InvocID), and was assigned the same process identification number (ElmtRunIndex) as an element which was previously run by the same invoked application.

Should this situation occur, the new entry replaces the old entry.

See the Implementation Issues section, sysAppElmtPastRunTable Entry Collisions for the conditions that would have to occur in order for a collision to occur.

::= { sysAppElmtPastRunEntry 2 }

**sysAppElmtPastRunInstallID OBJECT-TYPE**

SYNTAX Unsigned32 (1..'ffffff'h)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The index into the installed element table. The value of this object is the same value as the sysAppInstallElmtIndex for the application element of which this entry represents a previously executed process.

::= { sysAppElmtPastRunEntry 3 }

**sysAppElmtPastRunTimeStarted OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The time the process was started.

::= { sysAppElmtPastRunEntry 4 }

#### **sysAppElmtPastRunTimeEnded OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The time the process ended.

::= { sysAppElmtPastRunEntry 5 }

#### **sysAppElmtPastRunName OBJECT-TYPE**

SYNTAX LongUtf8String

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The full path and filename of the process. For example, '/opt/MYYpkg/bin/myyproc' would be returned for process 'myyproc' whose execution path was '/opt/MYYpkg/bin/myyproc'.

::= { sysAppElmtPastRunEntry 6 }

#### **sysAppElmtPastRunParameters OBJECT-TYPE**

SYNTAX Utf8String

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The starting parameters for the process.

::= { sysAppElmtPastRunEntry 7 }

#### **sysAppElmtPastRunCPU OBJECT-TYPE**

SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The last known number of centi-seconds of the total system's CPU resources consumed by this process. Note that on a multi-processor system, this value may increment by more than one centi-second in one centi-second of real (wall clock) time.

::= { sysAppElmtPastRunEntry 8 }

#### **sysAppElmtPastRunMemory OBJECT-TYPE**

SYNTAX Unsigned32 (0..'ffffff'h)

UNITSKbytes

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The last known total amount of real system memory measured in Kbytes allocated to this process before it terminated.

::= { sysAppElmtPastRunEntry 9 }

#### **sysAppElmtPastRunNumFiles OBJECT-TYPE**

SYNTAX Unsigned32 (0..'ffffff'h)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The last known number of files open by the process before it terminated. Transport connections (sockets) should NOT be included in the calculation of this value.

::= { sysAppElmtPastRunEntry 10 }

#### **sysAppElmtPastRunUser OBJECT-TYPE**

SYNTAX Utf8String

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The process owner's login name (e.g. root).

::= { sysAppElmtPastRunEntry 11 }

## テーブル サイズを制御する追加のスカラ オブジェクト

#### **sysApplPastRunMaxRows OBJECT-TYPE**

SYNTAX Unsigned32 (0..'ffffff'h)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

The maximum number of entries allowed in the sysApplPastRunTable. Once the number of rows in the sysApplPastRunTable reaches this value, the management subsystem will remove the oldest entry in the table to make room for the new entry to be added. Entries will be removed on the basis of oldest sysApplPastRunTimeEnded value first.

This object may be used to control the amount of system resources that can be used for sysApplPastRunTable entries. A conforming implementation should attempt to support the default value, however, a lesser value may be necessary due to implementation-dependent issues and resource availability.

DEFVAL { 500 }

::= { sysApplRun 5 }

**sysApplPastRunTableRemItems OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A counter of the number of entries removed from the sysApplPastRunTable because of table size limitations as set in sysApplPastRunMaxRows. This counter is the number of entries the management subsystem has had to remove in order to make room for new entries (so as not to exceed the limit set by sysApplPastRunMaxRows) since the last initialization of the management subsystem.

::= { sysApplRun 6 }

**sysApplPastRunTblTimeLimit OBJECT-TYPE**

SYNTAX Unsigned32 (0..'ffffff'h)

UNITSseconds

MAX-ACCESS read-write

STATUS current

DESCRIPTION

The maximum time in seconds which an entry in the sysApplPastRunTable may exist before it is removed. Any entry that is older than this value will be removed (aged out) from the table. Note that an entry may be aged out prior to reaching this time limit if it is the oldest entry in the table and must be removed to make space for a new entry so as to not exceed sysApplPastRunMaxRows.

DEFVAL { 7200 }

::= { sysApplRun 7 }

**sysApplElemPastRunMaxRows OBJECT-TYPE**

SYNTAX Unsigned32 (0..'ffffff'h)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

The maximum number of entries allowed in the sysApplElmtPastRunTable. Once the number of rows in the sysApplElmtPastRunTable reaches this value, the management subsystem will remove the oldest entry to make room for the new entry to be added. Entries will be removed on the basis of oldest sysApplElmtPastRunTimeEnded value first. This object may be used to control the amount of system resources that can be used for sysApplElemPastRunTable entries. A conforming implementation should attempt to support the default value, however, a lesser value may be necessary due to implementation-dependent issues and resource availability.

DEFVAL { 500 }

::= { sysApplRun 8 }

**sysApplElemPastRunTableRemItems OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

A counter of the number of entries removed from the sysAppElemPastRunTable because of table size limitations as set in sysAppElemPastRunMaxRows. This counter is the number of entries the management subsystem has had to remove in order to make room for new entries (so as not to exceed the limit set by sysAppElemPastRunMaxRows) since the last initialization of the management subsystem.

::= { sysApplRun 9 }

**sysAppElemPastRunTblTimeLimit OBJECT-TYPE**

SYNTAX Unsigned32 (0..'ffffff'h)

UNITS seconds

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

The maximum time in seconds which an entry in the sysAppElemPastRunTable may exist before it is removed. Any entry that is older than this value will be removed (aged out) from the table. Note that an entry may be aged out prior to reaching this time limit if it is the oldest entry in the table and must be removed to make space for a new entry so as to not exceed sysAppElemPastRunMaxRows.

DEFVAL { 7200 }

::= { sysApplRun 10 }

**sysAppAgentPollInterval OBJECT-TYPE**

SYNTAX Unsigned32 (0..'ffffff'h)

UNITS seconds

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

The minimum interval in seconds that the management subsystem implementing this MIB will poll the status of the managed resources. Because of the non-trivial effort involved in polling the managed resources, and because the method for obtaining the status of the managed resources is implementation-dependent, a conformant implementation may chose a lower bound greater than 0.

A value of 0 indicates that there is no delay in the passing of information from the managed resources to the agent.

DEFVAL { 60 }

::= { sysApplRun 11 }

## sysAppMap グループ

このグループには、テーブル sysAppMapTable が含まれます。このテーブルの唯一の目的は、「後方」マッピングを提供して、付与された既知の sysAppElmtRunIndex (プロセス識別番号)、対応する呼び出されたアプリケーション (sysAppRunIndex)、インストールされた要素 (sysAppInstallElmtIndex)、およびインストールされたアプリケーション パッケージ (sysAppInstallPkgIndex) をすばやく判断できるようにすることです。このテーブルには、システムで現在実行中の各プロセスの 1 つのエントリが含まれています。

この MIB モジュール内のテーブルは、通常インストールされたアプリケーション パッケージ (sysApplInstallPkgIndex) をプライマリ キーとして使用し、特定のテーブルにより必要とされるように重要性の最も低いキーであるプロセス ID 番号 (sysApplElmtRunIndex) を使用して下に向かってインデックスが付けられるため、後方マッピングは非常に役立ちます。

管理アプリケーションは、部分インスタンス ID として既知のプロセス ID 番号 (sysApplElmtRunIndex) を使用して「GetNext」オペレーションを実行することで、このマッピングテーブルを使用します。プロセスのエントリが 1 つ存在すると仮定すると、結果として 1 つのテーブル値 sysApplMapInstallPkgIndex が返され、返される MIB オブジェクト値のインスタンス ID には sysApplElmtRunIndex、sysApplRunIndex、および sysApplInstallElmtIndex が含まれます。



(注)

システムにインストールされた、呼び出されたアプリケーションにプロセスを再び関連付けることができない場合、テーブル値 sysApplMapInstallPkgIndex に返される値は「0」になり、オブジェクト ID のインスタンス部分は末尾に 0.0 が付くプロセス ID 番号 (sysApplElmtRunIndex) となります。

### sysApplMapTable OBJECT-TYPE

SYNTAX SEQUENCE OF SysApplMapEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The sole purpose of this table is to provide a 'backwards' mapping so that, given a known sysApplElmtRunIndex (process identification number), the corresponding invoked application (sysApplRunIndex), installed element (sysApplInstallElmtIndex), and installed application package (sysApplInstallPkgIndex) can be quickly determined.

::= { sysApplMap 1 }

### sysApplMapEntry OBJECT-TYPE

SYNTAX SysApplMapEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

A logical row representing a process currently running on the system. This entry provides the index mapping from process identifier, back to the invoked application, installed element, and finally, the installed application package. The entry includes only one accessible columnar object, the sysApplMapInstallPkgIndex, but the invoked application and installed element can be determined from the instance identifier since they form part of the index clause.

INDEX { sysApplElmtRunIndex, sysApplElmtRunInvocID, sysApplMapInstallElmtIndex }

::= { sysApplMapTable 1 }

SysApplMapEntry ::= SEQUENCE { sysApplMapInstallElmtIndexUnsigned32, sysApplMapInstallPkgIndex Unsigned32 }

### sysApplMapInstallElmtIndex OBJECT-TYPE

SYNTAX Unsigned32 (0..'ffffff'h)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION



The index into the sysApplInstallElmtTable. The value of this object is the same value as the sysApplInstallElmtIndex for the application element of which this entry represents a running instance. If this process cannot be associated to an installed executable, the value should be '0'.

::= { sysApplMapEntry 1 }

#### **sysApplMapInstallPkgIndex OBJECT-TYPE**

SYNTAX Unsigned32 (0..'ffffff'h)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The value of this object identifies the installed software package for the application of which this process is a part. Provided that the process's 'parent' application can be determined, the value of this object is the same value as the sysApplInstallPkgIndex for the entry in the sysApplInstallPkgTable that corresponds to the installed application of which this process is a part.

If, however, the 'parent' application cannot be determined, (for example the process is not part of a particular installed application), the value for this object is then '0', signifying that this process cannot be related back to an application, and in turn, an installed software package.

::= { sysApplMapEntry 2 }

## 適合マクロ

**sysApplMIBCompliances OBJECT IDENTIFIER ::= { sysApplConformance 1 }**

**sysApplMIBGroups OBJECT IDENTIFIER ::= { sysApplConformance 2 }**

#### **sysApplMIBCompliance MODULE-COMPLIANCE**

STATUS current

DESCRIPTION

Describes the requirements for conformance to the System Application MIB MODULE.

MANDATORY-GROUPS { sysApplInstalledGroup, sysApplRunGroup, sysApplMapGroup }

::= { sysApplMIBCompliances 1 }

#### **sysApplInstalledGroup OBJECT-GROUP**

OBJECTS { sysApplInstallPkgManufacturer, sysApplInstallPkgProductName, sysApplInstallPkgVersion, sysApplInstallPkgSerialNumber, sysApplInstallPkgDate, sysApplInstallPkgLocation, sysApplInstallElmtName, sysApplInstallElmtType, sysApplInstallElmtDate, sysApplInstallElmtPath, sysApplInstallElmtSizeHigh, sysApplInstallElmtSizeLow, sysApplInstallElmtRole, sysApplInstallElmtModifyDate, sysApplInstallElmtCurSizeHigh, sysApplInstallElmtCurSizeLow }

STATUS current

DESCRIPTION

The system application installed group contains information about applications and their constituent components which have been installed on the host system.

::= { sysApplMIBGroups 1 }

#### **sysApplRunGroup OBJECT-GROUP**

OBJECTS { sysApplRunStarted, sysApplRunCurrentState, sysApplPastRunStarted, sysApplPastRunExitState, sysApplPastRunTimeEnded, sysApplElmtRunInstallID, sysApplElmtRunTimeStarted, sysApplElmtRunState, sysApplElmtRunName, sysApplElmtRunParameters, sysApplElmtRunCPU, sysApplElmtRunMemory, sysApplElmtRunNumFiles, sysApplElmtRunUser, sysApplElmtPastRunInstallID, sysApplElmtPastRunTimeStarted, sysApplElmtPastRunTimeEnded, sysApplElmtPastRunName, sysApplElmtPastRunParameters, sysApplElmtPastRunCPU, sysApplElmtPastRunMemory, sysApplElmtPastRunNumFiles, sysApplElmtPastRunUser, sysApplPastRunMaxRows, sysApplPastRunTableRemItems, sysApplPastRunTblTimeLimit, sysApplElemPastRunMaxRows, sysApplElemPastRunTableRemItems, sysApplElemPastRunTblTimeLimit, sysApplAgentPollInterval }

STATUS current

DESCRIPTION

The system application run group contains information about applications and associated elements which have run or are currently running on the host system.

::= { sysApplMIBGroups 2 }

#### sysApplMapGroup OBJECT-GROUP

OBJECTS { sysApplMapInstallPkgIndex }

STATUS current

DESCRIPTION

The Map Group contains a single table, sysApplMapTable, that provides a backwards mapping for determining the invoked application, installed element, and installed application package given a known process identification number.

::= { sysApplMIBGroups 3 }

## トラブルシューティング

次の各項に、トラブルシューティングのヒントを示します。

- 「Linux および Cisco Unified CM Release 5.x、6.x、7.x」 (P.8-26)
- 「Windows および Cisco Unified CM Release 4.x」 (P.8-27)
- 「Cisco Unified CM 7.x でのサブプレットの使用」 (P.8-27)
- 「FAQ」 (P.8-28)

## Linux および Cisco Unified CM Release 5.x、6.x、7.x

次のログおよび情報を収集して分析します。コマンド **file get activelog <paths below>** を実行します。

- SNMP マスター エージェントのパス : /platform/snmp/snmpdm/\*
- システム アプリケーション エージェントのパス : /platform/snmp/sappagt/\*

## Windows および Cisco Unified CM Release 4.x

次のログおよび情報を収集して分析します。

- レジストリ HKEY\_LOCAL\_MACHINE¥SOFTWARE¥Cisco Systems, Inc.¥SnmpSysAppAgent の Enable TraceEnabled を「true」に、TraceLevel を「3」にして、sysapp トレース レベルを「詳細」に設定します。
- 編集したら、[ サービス (Services) ] タブから SNMP サービスを再起動します。トレース ファイル C:¥Program Files¥Cisco¥bin¥SnmpSysAppImpl.log が作成されます。
- sysAppInstallPkgTable で snmpwalk を実行します。
- SysAppRunTable で snmpwalk を実行します。
- ウォークが完了したら、C:¥Program Files¥Cisco¥bin¥SnmpSysAppImpl.log ログ ファイルを収集します。
- イベント ログ ビューアからアプリケーション ログとイベント ログを収集します。

## Cisco Unified CM 7.x でのサーブレットの使用

SysApp MIB には、インストールされて実行されているコンポーネントを一度に取得する方法が用意されています。SysApp エージェントは、アクティブまたは非アクティブになっているサービスのリストを表示できません。アプリケーションとサービスの実行中状態または実行中以外の状態だけ示すことができます。Web アプリケーションのサービスやサーブレットは、SysApp MIB を使用してモニタできません。7.x システムのサーブレットは次のとおりです。

- Cisco CallManager Admin
- Cisco CallManager Cisco IP Phone Service
- Cisco CallManager Personal Directory
- Cisco CallManager Serviceability
- Cisco CallManager Serviceability RTMT
- Cisco Dialed Number Analyzer
- Cisco エクステンション モビリティ
- Cisco エクステンション モビリティ アプリケーション
- Cisco RTMT Reporter Servlet
- Cisco Tomcat Stats Servlet
- Cisco Trace Collection Servlet
- Cisco AXL Web Service
- Cisco Unified Mobile Voice Access Service
- Cisco エクステンション モビリティ
- Cisco IP Manager Assistant
- Cisco WebDialer Web Service
- Cisco CAR Web Service
- Cisco Dialed Number Analyzer

システムの健全性の目的で重要なサービスのステータスをモニタする場合は、次の方法を推奨します。

- `GetServiceStatus` と呼ばれる `Serviceability API` を使用します。この API は、Web アプリケーション タイプと非 Web アプリケーション サービスの両方の完全なステータス情報（アクティベーション ステータスなど）を取得できます（詳細については、『*AXL Serviceability API Guide*』を参照してください）。
- `utils service list` コマンドを使用して、さまざまなサービスのステータスをチェックします。
- `syslog` メッセージを使用して、`servM` で生成されたメッセージをモニタします。次に例を示します。

```
Mar 18 16:40:52 ciscart26 local17 6 : 92: Mar 18 11:10:52.630 UTC :
%CCM_SERVICEMANAGER-SERVICEMANAGER-6-ServiceActivated: Service Activated. Service
Name: Cisco CallManager SNMP Service App ID: Cisco Service Manager Cluster ID: Node
ID:ciscart26
```

## FAQ

Cisco Unified CM Release 4.x で `CCMVersion MIB` と `sysAppRunCurrentState` が間違った値を返した場合は、`CSCsk74156` を参照して該当するかどうかをチェックしてください。使用している Cisco Unified CM のバージョン用の修正が提供されているかどうかを確認します。

`sysApp MIB` での SNMP ウォークが反応しない場合、`CSCsh72473` を参照して該当するかどうかをチェックしてください。使用している Cisco Unified CM のバージョン用の修正が提供されているかどうかを確認します。

## RFC1213-MIB (MIB-II)



(注)

この MIB-II は形式が変更されています。この項のすべての MIB は、<http://tools.cisco.com/Support/SNMP/do/BrowseMIB.do?local=en&step=2> からダウンロードしてコンパイルしてください。

RFC1213-MIB をコンパイルする前に、次のリストの MIB を、リストされている順にコンパイルする必要があります。

1. SNMPv2-SMI
2. SNMPv2-TC
3. IANAifType-MIB
4. RFC1155-SMI
5. RFC-1212
6. RFC1213-MIB

この項の内容は次のとおりです。

- 「改訂」 (P.8-29)
- 「定義」 (P.8-29)
- 「オブジェクト ID」 (P.8-29)
- 「テキストの表記法」 (P.8-29)
- 「MIB-II におけるグループ」 (P.8-30)

- 「従来のグループ」 (P.8-30)
- 「システム グループ」 (P.8-30)
- 「インターフェイス グループ」 (P.8-32)
- 「アドレス変換グループ」 (P.8-38)
- 「IP グループ」 (P.8-39)
- 「ICMP グループ」 (P.8-51)
- 「TCP グループ」 (P.8-56)
- 「UDP グループ」 (P.8-61)
- 「EGP グループ」 (P.8-62)
- 「SNMP グループ」 (P.8-67)

## 改訂

次の改訂が適用されました。

- RFC 1573 で導入された ifTable の変更を反映するため、列挙 unknown(4) および dormant(5) が ifOperStatus に追加されました。
- RFC1573 で導入された ifTable の変更を反映するため、ifType の SYNTAX が IANAifType に変更されました。

## 定義

次の定義が MIB-II 用にインポートされています。

- mgmt、NetworkAddress、IpAddress、Counter、Gauge、TimeTicks
- RFC1155-SMI から : OBJECT-TYPE
- RFC-1212 から : TEXTUAL-CONVENTION
- SNMPv2-TC から : IANAifType
- IANAifType-MIB から

## オブジェクト ID

この MIB モジュールは、[14] で定義されているように、拡張 OBJECT-TYPE マクロを使用します。MIB-II (MIB-I と同じプレフィクス) mib-2 OBJECT IDENTIFIER ::= { mgmt 1 }。

## テキストの表記法

**DisplayString ::= OCTET STRING**

このデータ タイプは、NVT ASCII 文字セットから取得されたテキスト情報のモデル化に使用されません。標準では、この構文のオブジェクトは SIZE (0..255) を持つと宣言されます。

**PhysAddress ::= OCTET STRING**

このデータタイプは、メディアアドレスのモデル化に使用されます。多くのタイプのメディアでは、これはバイナリ表現となります。たとえば、イーサネットアドレスは 6 オクテットの文字列で表現されます。

## MIB-II におけるグループ

system	OBJECT IDENTIFIER ::= { mib-2 1 }
interfaces	OBJECT IDENTIFIER ::= { mib-2 2 }
at	OBJECT IDENTIFIER ::= { mib-2 3 }
ip	OBJECT IDENTIFIER ::= { mib-2 4 }
icmp	OBJECT IDENTIFIER ::= { mib-2 5 }
tcp	OBJECT IDENTIFIER ::= { mib-2 6 }
udp	OBJECT IDENTIFIER ::= { mib-2 7 }
egp	OBJECT IDENTIFIER ::= { mib-2 8 }

## 従来のグループ

cmot	OBJECT IDENTIFIER ::= { mib-2 9 }
transmission	OBJECT IDENTIFIER ::= { mib-2 10 }
snmp	OBJECT IDENTIFIER ::= { mib-2 11 }

## システム グループ

システム グループの実装は、すべてのシステムで必須です。これらの変数の値を持つようにエージェントが設定されていない場合、長さ 0 の文字列が返されます。

### sysDescr OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..255))

ACCESS read-only

STATUS mandatory

DESCRIPTION

A textual description of the entity. This value should include the full name and version identification of the system's hardware type, software operating-system, and networking software. It is mandatory that this only contain printable ASCII characters.

::= { system 1 }

### sysObjectID OBJECT-TYPE

SYNTAX Object Identifier

ACCESS read-only

STATUS mandatory

DESCRIPTION

The vendor authoritative identification of the network management subsystem contained in the entity. This value is allocated within the SMI enterprises subtree (1.3.6.1.4.1) and provides an easy and unambiguous means for determining "what kind of box" is being managed. For example, if vendor "Flintstones, Inc." was assigned the subtree 1.3.6.1.4.1.4242, it could assign the identifier 1.3.6.1.4.1.4242.1.1 to its "Fred Router".

::= { system 2 }

**sysUpTime OBJECT-TYPE**

SYNTAX TimeTicks

ACCESS read-only

STATUS mandatory

DESCRIPTION

The time (in hundredths of a second) since the network management portion of the system was last re-initialized.

::= { system 3 }

**sysContact OBJECT-TYPE**

SYNTAX DisplayString (SIZE (0..255))

ACCESS read-write

STATUS mandatory

DESCRIPTION

The textual identification of the contact person for this managed node, together with information on how to contact this person.

::= { system 4 }

**sysName OBJECT-TYPE**

SYNTAX DisplayString (SIZE (0..255))

ACCESS read-write

STATUS mandatory

DESCRIPTION

An administratively-assigned name for this managed node. By convention, this is the node's fully-qualified domain name.

::= { system 5 }

**sysLocation OBJECT-TYPE**

SYNTAX DisplayString (SIZE (0..255))

ACCESS read-write

STATUS mandatory

DESCRIPTION

The physical location of this node (e.g., telephone closet, 3rd floor).

::= { system 6 }

**sysServices OBJECT-TYPE**

SYNTAX Integer (0..127)

ACCESS read-only

STATUS mandatory

DESCRIPTION

A value which indicates the set of services that this entity primarily offers. The value is a sum. This sum initially takes the value zero. Then, for each layer, L, in the range 1 through 7, that this node performs transactions for,  $2$  raised to  $(L - 1)$  is added to the sum. For example, a node which performs primarily routing functions would have a value of  $4$  ( $2^{(3-1)}$ ). In contrast, a node which is a host offering application services would have a value of  $72$  ( $2^{(4-1)} + 2^{(7-1)}$ ). Note that in the context of the Internet suite of protocols, values should be calculated accordingly (layer first, then functionality):

1 physical (e.g., repeaters)

2 datalink/subnetwork (e.g., bridges)

3 internet (e.g., IP gateways)

4 end-to-end (e.g., IP hosts)

7 applications (e.g., mail relays)

For systems including OSI protocols, layers 5 and 6 may also be counted.

::= { system 7 }

## インターフェイス グループ

インターフェイス グループの実装は、すべてのシステムで必須です。

**ifNumber OBJECT-TYPE**

SYNTAX Integer

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of network interfaces (regardless of their current state) present on this system.

::= { interfaces 1 }

## インターフェイス テーブル

インターフェイス テーブルには、エンティティ インターフェイスに関する情報が格納されています。各インターフェイスは、サブネットワークに接続されていると考えられます。この用語を、プロトコルのインターネットスイートで使用されるアドレス パーティション指定方式を指すサブネットワークと混同しないでください。

**ifTable OBJECT-TYPE**

SYNTAX Sequence of ifEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

A list of interface entries. The number of entries is given by the value of ifNumber.

::= { interfaces 2 }



**ifEntry OBJECT-TYPE**

SYNTAX IfEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

An interface entry containing objects at the subnetwork layer and below for a particular interface.

INDEX { ifIndex }

::= { ifTable 1 }

IfEntry ::=

SEQUENCE { ifIndex INTEGER, ifDescr DisplayString, ifType IANAifType, ifMtu INTEGER, ifSpeed Gauge, ifPhysAddress PhysAddress, ifAdminStatus INTEGER, ifOperStatus INTEGER, ifLastChange TimeTicks, ifInOctets Counter, ifInUcastPkts Counter, ifInNUcastPkts Counter, ifInDiscards Counter, ifInErrors Counter, ifInUnknownProtos Counter, ifOutOctets Counter, ifOutUcastPkts Counter, ifOutNUcastPkts Counter, ifOutDiscards Counter, ifOutErrors Counter, ifOutQLen Gauge, ifSpecific OBJECT IDENTIFIER }

**ifIndex OBJECT-TYPE**

SYNTAX Integer

ACCESS read-only

STATUS mandatory

DESCRIPTION

A unique value for each interface. Its value ranges between 1 and the value of ifNumber. The value for each interface must remain constant at least from one re-initialization of the entity network management system to the next re-initialization.

::= { ifEntry 1 }

**ifDescr OBJECT-TYPE**

SYNTAX DisplayString (SIZE (0..255))

ACCESS read-only

STATUS mandatory

DESCRIPTION

A textual string containing information about the interface. This string should include the name of the manufacturer, the product name and the version of the hardware interface.

::= { ifEntry 2 }

**ifType OBJECT-TYPE**

SYNTAX IANAifType

ACCESS read-only

STATUS mandatory

DESCRIPTION

The type of interface. Additional values for ifType are assigned by the Internet Assigned Numbers Authority (IANA), through updating the syntax of the IANAifType textual convention.

::= { ifEntry 3 }

**ifMtu OBJECT-TYPE**

SYNTAX Integer

ACCESS read-only

STATUS mandatory

DESCRIPTION

The size of the largest datagram which can be sent/received on the interface, specified in octets. For interfaces that are used for transmitting network datagrams, this is the size of the largest network datagram that can be sent on the interface.

::= { ifEntry 4 }

**ifSpeed OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

An estimate of the interface current bandwidth in bits per second. For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth.

::= { ifEntry 5 }

**ifPhysAddress OBJECT-TYPE**

SYNTAX PhysAddress

ACCESS read-only

STATUS mandatory

DESCRIPTION

The interface address at the protocol layer immediately below the network layer in the protocol stack. For interfaces which do not have such an address (e.g., a serial line), this object should contain an octet string of zero length.

::= { ifEntry 6 }

**ifAdminStatus OBJECT-TYPE**

SYNTAX Integer { up(1), ready to pass packets down(2), testing(3) in some test mode }

ACCESS read-write

STATUS mandatory

DESCRIPTION

The desired state of the interface. The testing(3) state indicates that no operational packets can be passed.

::= { ifEntry 7 }

**ifOperStatus OBJECT-TYPE**SYNTAX INTEGER { up(1), -- ready to pass packets down(2), testing(3), -- in some test mode  
unknown(4), dormant(5) }

ACCESS read-only

STATUS mandatory

## DESCRIPTION

The current operational state of the interface. The testing(3) state indicates that no operational packets can be passed.

::= { ifEntry 8 }

**ifLastChange OBJECT-TYPE**

SYNTAX TimeTicks

ACCESS read-only

STATUS mandatory

## DESCRIPTION

The value of sysUpTime at the time the interface entered its current operational state. If the current state was entered prior to the last re-initialization of the local network management subsystem, then this object contains a zero value.

::= { ifEntry 9 }

**ifInOctets OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION The total number of octets received on the interface, including framing characters.

::= { ifEntry 10 }

**ifInUcastPkts OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

## DESCRIPTION

The number of subnetwork-unicast packets delivered to a higher-layer protocol.

::= { ifEntry 11 }

**ifInNUcastPkts OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

## DESCRIPTION

The number of non-unicast (i.e., subnetwork-broadcast or subnetwork-multicast) packets delivered to a higher-layer protocol.

::= { ifEntry 12 }

**ifInDiscards OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

## DESCRIPTION

The number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space.

::= { ifEntry 13 }

#### **ifInErrors OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.

::= { ifEntry 14 }

#### **ifInUnknownProtos OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of packets received via the interface which were discarded because of an unknown or unsupported protocol.

::= { ifEntry 15 }

#### **ifOutOctets OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of octets transmitted out of the interface, including framing characters.

::= { ifEntry 16 }

#### **ifOutUcastPkts OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of packets that higher-level protocols requested be transmitted to a subnetwork-unicast address, including those that were discarded or not sent.

::= { ifEntry 17 }

#### **ifOutNUcastPkts OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

## DESCRIPTION

The total number of packets that higher-level protocols requested be transmitted to a non-unicast (i.e., a subnetwork-broadcast or subnetwork-multicast) address, including those that were discarded or not sent.

::= { ifEntry 18 }

**ifOutDiscards OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

## DESCRIPTION

The number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space.

::= { ifEntry 19 }

**ifOutErrors OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

## DESCRIPTION

The number of outbound packets that could not be transmitted because of errors.

::= { ifEntry 20 }

**ifOutQLen OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

## DESCRIPTION

The length of the output packet queue (in packets).

::= { ifEntry 21 }

**ifSpecific OBJECT-TYPE**

SYNTAX OBJECT IDENTIFIER

ACCESS read-only

STATUS mandatory

## DESCRIPTION

A reference to MIB definitions specific to the particular media being used to realize the interface. For example, if the interface is realized by an ethernet, then the value of this object refers to a document defining objects specific to ethernet. If this information is not present, its value should be set to the OBJECT IDENTIFIER { 0 0 }, which is a syntactically valid object identifier, and any conformant implementation of ASN.1 and BER must be able to generate and recognize this value.

::= { ifEntry 22 }

## アドレス変換グループ

アドレス変換グループの実装は、すべてのシステムで必須です。ただし、このグループは MIB-II で非推奨である点に注意してください。つまり、MIB-I ノードとの互換性のためだけに用意されており、MIB-III ノードでは除外される予定です。MIB-II 以降では、各ネットワーク プロトコル グループにその独自のアドレス変換テーブルが含まれます。アドレス変換グループには、NetworkAddress (IP アドレスなど) をサブネットワーク固有のアドレスに変換するための、すべてのインターフェイスにおける変換テーブルの結合である 1 つのテーブルが含まれます。適切な用語がないため、このマニュアルではそのようなサブネットワーク固有のアドレスを物理アドレスと呼ぶことにします。

そのような変換テーブルの例として、Address Resolution Protocol (ARP; アドレス解決プロトコル) を使用しているブロードキャスト メディアの場合、変換テーブルは ARP キャッシュに相当します。または X.121 アドレスへの非アルゴリズム変換が必要な X.25 ネットワークでは、同等の X.121 アドレスに対する NetworkAddress が変換テーブルに格納されます。

### atTable OBJECT-TYPE

SYNTAX Sequence of atEntry

ACCESS not-accessible

STATUS deprecated

DESCRIPTION

The Address Translation tables contain the NetworkAddress to physical address equivalences. Some interfaces do not use translation tables for determining address equivalences (e.g., DDN-X.25 has an algorithmic method); if all interfaces are of this type, then the Address Translation table is empty, i.e., has zero entries.

::= { at 1 }

### atEntry OBJECT-TYPE

SYNTAX AtEntry

ACCESS not-accessible

STATUS deprecated

DESCRIPTION

Each entry contains one NetworkAddress to physical address equivalence.

INDEX { atIfIndex, atNetAddress }

::= { atTable 1 }

AtEntry ::=

SEQUENCE { atIfIndex INTEGER, atPhysAddress PhysAddress, atNetAddress NetworkAddress }

### atIfIndex OBJECT-TYPE

SYNTAX Integer

ACCESS read-write

STATUS deprecated

DESCRIPTION

The interface on which this entry equivalence is effective. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

::= { atEntry 1 }

### atPhysAddress OBJECT-TYPE

SYNTAX PhysAddress

ACCESS read-write

STATUS deprecated

DESCRIPTION

The media-dependent physical address. Setting this object to a null string (one of zero length) has the effect of invalidating the corresponding entry in the atTable object. That is, it effectively disassociates the interface identified with said entry from the mapping identified with said entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use.

Proper interpretation of such entries requires examination of the relevant atPhysAddress object.

::= { atEntry 2 }

#### **atNetAddress OBJECT-TYPE**

SYNTAX NetworkAddress

ACCESS read-write

STATUS deprecated

DESCRIPTION

The NetworkAddress (e.g., the IP address) corresponding to the media-dependent physical address.

::= { atEntry 3 }

## IP グループ

IP グループの実装は、すべてのシステムで必須です。

#### **ipForwarding OBJECT-TYPE**

SYNTAX INTEGER { forwarding(1), -- acting as a gateway not-forwarding(2) -- NOT acting as a gateway }

ACCESS read-write

STATUS mandatory

DESCRIPTION

The indication of whether this entity is acting as an IP gateway in respect to the forwarding of datagrams received by, but not addressed to, this entity. IP gateways forward datagrams. IP hosts do not (except those source-routed via the host). Note that for some managed nodes, this object may take on only a subset of the values possible. Accordingly, it is appropriate for an agent to return a badValue response if a management station attempts to change this object to an inappropriate value.

::= { ip 1 }

#### **ipDefaultTTL OBJECT-TYPE**

SYNTAX Integer

ACCESS read-write

STATUS mandatory

DESCRIPTION

The default value inserted into the Time-To-Live field of the IP header of datagrams originated at this entity, whenever a TTL value is not supplied by the transport layer protocol.

::= { ip 2 }

#### **ipInReceives OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of input datagrams received from interfaces, including those received in error.

::= { ip 3 }

#### **ipInHdrErrors OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of input datagrams discarded due to errors in their IP headers, including bad checksums, version number mismatch, other format errors, time-to-live exceeded, errors discovered in processing their IP options, etc.

::= { ip 4 }

#### **ipInAddrErrors OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of input datagrams discarded because the IP address in their IP header's destination field was not a valid address to be received at this entity. This count includes invalid addresses (e.g., 0.0.0.0) and addresses of unsupported Classes (e.g., Class E). For entities which are not IP Gateways and therefore do not forward datagrams, this counter includes datagrams discarded because the destination address was not a local address.

::= { ip 5 }

#### **ipForwDatagrams OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of input datagrams for which this entity was not their final IP destination, as a result of which an attempt was made to find a route to forward them to that final destination. In entities which do not act as IP Gateways, this counter will include only those packets which were Source-Routed via this entity, and the Source-Route option processing was successful.

::= { ip 6 }



**ipInUnknownProtos OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of locally-addressed datagrams received successfully but discarded because of an unknown or unsupported protocol.

::= { ip 7 }

**ipInDiscards OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of input IP datagrams for which no problems were encountered to prevent their continued processing, but which were discarded (e.g., for lack of buffer space). Note that this counter does not include any datagrams discarded while awaiting re-assembly.

::= { ip 8 }

**ipInDelivers OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of input datagrams successfully delivered to IP user-protocols (including ICMP).

::= { ip 9 }

**ipOutRequests OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of IP datagrams which local IP user-protocols (including ICMP) supplied to IP in requests for transmission. Note that this counter does not include any datagrams counted in ipForwDatagrams.

::= { ip 10 }

**ipOutDiscards OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of output IP datagrams for which no problem was encountered to prevent their transmission to their destination, but which were discarded (e.g., for lack of buffer space). Note that this counter would include datagrams counted in ipForwDatagrams if any such packets met this (discretionary) discard criterion.

::= { ip 11 }

#### **ipOutNoRoutes OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of IP datagrams discarded because no route could be found to transmit them to their destination. Note that this counter includes any packets counted in ipForwDatagrams which meet this no-route criterion. Note that this includes any datagrams which a host cannot route because all of its default gateways are down.

::= { ip 12 }

#### **ipReasmTimeout OBJECT-TYPE**

SYNTAX Integer

ACCESS read-only

STATUS mandatory

DESCRIPTION

The maximum number of seconds which received fragments are held while they are awaiting reassembly at this entity.

::= { ip 13 }

#### **ipReasmReqds OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of IP fragments received which needed to be reassembled at this entity.

::= { ip 14 }

#### **ipReasmOKs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of IP datagrams successfully re-assembled.

::= { ip 15 }

#### **ipReasmFails OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of failures detected by the IP re-assembly algorithm (for whatever reason: timed out, errors, etc). Note that this is not necessarily a count of discarded IP fragments since some algorithms (notably the algorithm in RFC 815) can lose track of the number of fragments by combining them as they are received.

::= { ip 16 }

#### **ipFragOKs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of IP datagrams that have been successfully fragmented at this entity.

::= { ip 17 }

#### **ipFragFails OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of IP datagrams that have been discarded because they needed to be fragmented at this entity but could not be, e.g., because their Don't Fragment flag was set.

::= { ip 18 }

#### **ipFragCreates OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of IP datagram fragments that have been generated as a result of fragmentation at this entity.

::= { ip 19 }

## IP アドレス テーブル

IP アドレス テーブルには、次のエンティティ IP アドレス指定情報が格納されています。

#### **ipAddrTable OBJECT-TYPE**

SYNTAX Sequence of ipAddrEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

The table of addressing information relevant to this entity IP addresses.

::= { ip 20 }

**ipAddrEntry OBJECT-TYPE**

SYNTAX IpAddrEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

The addressing information for one of this entity IP addresses.

INDEX { ipAdEntAddr }

::= { ipAddrTable 1 }

IpAddrEntry ::=

SEQUENCE { ipAdEntAddr IpAddress, ipAdEntIfIndex INTEGER, ipAdEntNetMask IpAddress, ipAdEntBcastAddr INTEGER, ipAdEntReasmMaxSize INTEGER (0..65535) }

**ipAdEntAddr OBJECT-TYPE**

SYNTAX IpAddress

ACCESS read-only

STATUS mandatory

DESCRIPTION

The IP address to which this entry addressing information pertains.

::= { ipAddrEntry 1 }

**ipAdEntIfIndex OBJECT-TYPE**

SYNTAX Integer

ACCESS read-only

STATUS mandatory

DESCRIPTION

The index value which uniquely identifies the interface to which this entry is applicable. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

::= { ipAddrEntry 2 }

**ipAdEntNetMask OBJECT-TYPE**

SYNTAX IpAddress

ACCESS read-only

STATUS mandatory

DESCRIPTION

The subnet mask associated with the IP address of this entry. The value of the mask is an IP address with all the network bits set to 1 and all the hosts bits set to 0.

::= { ipAddrEntry 3 }

**ipAdEntBcastAddr OBJECT-TYPE**

SYNTAX Integer

ACCESS read-only

STATUS mandatory

DESCRIPTION

The value of the least-significant bit in the IP broadcast address used for sending datagrams on the (logical) interface associated with the IP address of this entry. For example, when the Internet standard all-ones broadcast address is used, the value will be 1. This value applies to both the subnet and network broadcasts addresses used by the entity on this (logical) interface.

::= { ipAddrEntry 4 }

#### **ipAdEntReasmMaxSize OBJECT-TYPE**

SYNTAX Integer (0..65535)

ACCESS read-only

STATUS mandatory

DESCRIPTION

The size of the largest IP datagram which this entity can re-assemble from incoming IP fragmented datagrams received on this interface.

::= { ipAddrEntry 5 }

## IP ルーティング テーブル

-- IP ルーティング テーブルには、このエンティティが現在認識している

-- 各ルートのエントリが格納されています。

#### **ipRouteTable OBJECT-TYPE**

SYNTAX Sequence of ipRouteEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

This entity IP Routing table.

::= { ip 21 }

#### **ipRouteEntry OBJECT-TYPE**

SYNTAX IpRouteEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

A route to a particular destination.

INDEX { ipRouteDest }

::= { ipRouteTable 1 }

IpRouteEntry ::=

SEQUENCE { ipRouteDest IpAddress, ipRouteIfIndex INTEGER, ipRouteMetric1 INTEGER, ipRouteMetric2 INTEGER, ipRouteMetric3 INTEGER, ipRouteMetric4 INTEGER, ipRouteNextHop IpAddress, ipRouteType INTEGER, ipRouteProto INTEGER, ipRouteAge INTEGER, ipRouteMask IpAddress, ipRouteMetric5 INTEGER, ipRouteInfo OBJECT IDENTIFIER }

**ipRouteDest OBJECT-TYPE**

SYNTAX IPAddress

ACCESS read-write

STATUS mandatory

DESCRIPTION

The destination IP address of this route. An entry with a value of 0.0.0.0 is considered a default route. Multiple routes to a single destination can appear in the table, but access to such multiple entries is dependent on the table-access mechanisms defined by the network management protocol in use.

::= { ipRouteEntry 1 }

**ipRouteIfIndex OBJECT-TYPE**

SYNTAX Integer

ACCESS read-write

STATUS mandatory

DESCRIPTION

The index value which uniquely identifies the local interface through which the next hop of this route should be reached. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

::= { ipRouteEntry 2 }

**ipRouteMetric1 OBJECT-TYPE**

SYNTAX Integer

ACCESS read-write

STATUS mandatory

DESCRIPTION

The primary routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route ipRouteProto value. If this metric is not used, its value should be set to -1.

::= { ipRouteEntry 3 }

**ipRouteMetric2 OBJECT-TYPE**

SYNTAX Integer

ACCESS read-write

STATUS mandatory

DESCRIPTION

An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route ipRouteProto value. If this metric is not used, its value should be set to -1.

::= { ipRouteEntry 4 }

**ipRouteMetric3 OBJECT-TYPE**

SYNTAX Integer

ACCESS read-write

STATUS mandatory

DESCRIPTION

An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route ipRouteProto value. If this metric is not used, its value should be set to -1.

::= { ipRouteEntry 5 }

#### **ipRouteMetric4 OBJECT-TYPE**

SYNTAX Integer

ACCESS read-write

STATUS mandatory

DESCRIPTION

An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route ipRouteProto value. If this metric is not used, its value should be set to -1.

::= { ipRouteEntry 6 }

#### **ipRouteNextHop OBJECT-TYPE**

SYNTAX IpAddress

ACCESS read-write

STATUS mandatory

DESCRIPTION

The IP address of the next hop of this route. (In the case of a route bound to an interface which is realized via a broadcast media, the value of this field is the agent's IP address on that interface.)

::= { ipRouteEntry 7 }

#### **ipRouteType OBJECT-TYPE**

SYNTAX Integer { other(1), -- none of the following invalid(2), -- an invalidated route --  
route to directly direct(3), -- connected (sub-)network -- route to a non-local indirect(4) --  
host/network/sub-network }

ACCESS read-write

STATUS mandatory

DESCRIPTION

The type of route. Note that the values direct(3) and indirect(4) refer to the notion of direct and indirect routing in the IP architecture. Setting this object to the value invalid(2) has the effect of invalidating the corresponding entry in the ipRouteTable object. That is, it effectively disassociates the destination identified with said entry from the route identified with said entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table.

Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant ipRouteType object.

::= { ipRouteEntry 8 }

#### **ipRouteProto OBJECT-TYPE**

SYNTAX INTEGER { other(1), -- none of the following -- non-protocol information, -- e.g., manually configured local(2), -- entries -- set via a network netmgmt(3), -- management protocol -- obtained via ICMP, icmp(4), -- e.g., Redirect -- the remaining values are -- all gateway routing -- protocols egp(5), ggp(6), hello(7), rip(8), is-is(9), es-is(10), ciscoIgrp(11), bbnSpflgp(12), ospf(13), bgp(14) }

ACCESS read-only

STATUS mandatory

DESCRIPTION

The routing mechanism via which this route was learned. Inclusion of values for gateway routing protocols is not intended to imply that hosts should support those protocols.

::= { ipRouteEntry 9 }

#### **ipRouteAge OBJECT-TYPE**

SYNTAX Integer

ACCESS read-write

STATUS mandatory

DESCRIPTION

The number of seconds since this route was last updated or otherwise determined to be correct. Note that no semantics of too old can be implied except through knowledge of the routing protocol by which the route was learned.

::= { ipRouteEntry 10 }

#### **ipRouteMask OBJECT-TYPE**

SYNTAX IPAddress

ACCESS read-write

STATUS mandatory

DESCRIPTION

Indicate the mask to be logical-ANDed with the destination address before being compared to the value in the ipRouteDest field. For those systems that do not support arbitrary subnet masks, an agent constructs the value of the ipRouteMask by determining whether the value of the correspondent ipRouteDest field belong to a class-A, B, or C network, and then using one of: mask network 255.0.0.0 class-A, 255.255.0.0 class-B, 255.255.255.0 class-C. If the value of the ipRouteDest is 0.0.0.0 (a default route), then the mask value is also 0.0.0.0. It should be noted that all IP routing subsystems implicitly use this mechanism.

::= { ipRouteEntry 11 }

#### **ipRouteMetric5 OBJECT-TYPE**

SYNTAX Integer

ACCESS read-write

STATUS mandatory

DESCRIPTION

An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route ipRouteProto value. If this metric is not used, its value should be set to -1.

::= { ipRouteEntry 12 }



**ipRouteInfo OBJECT-TYPE**

SYNTAX OBJECT IDENTIFIER

ACCESS read-only

STATUS mandatory

DESCRIPTION

A reference to MIB definitions specific to the particular routing protocol which is responsible for this route, as determined by the value specified in the route ipRouteProto value. If this information is not present, its value should be set to the OBJECT IDENTIFIER { 0 0 }, which is a syntactically valid object identifier, and any conformant implementation of ASN.1 and BER must be able to generate and recognize this value.

::= { ipRouteEntry 13 }

**IP アドレス変換テーブル**

IP アドレス変換テーブルには、同等の物理アドレスに対する IP アドレスが格納されています。一部のインターフェイスでは、同等のアドレスを判断するために変換テーブルを使用しません（たとえば、DDN-X.25 にはアルゴリズム方式があります）。すべてのインターフェイスがこのタイプである場合、アドレス変換テーブルは空、つまりエントリの数が 0 個です。

**ipNetToMediaTable OBJECT-TYPE**

SYNTAX Sequence of ipNetToMediaEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

The IP Address Translation table used for mapping from IP addresses to physical addresses.

::= { ip 22 }

**ipNetToMediaEntry OBJECT-TYPE**

SYNTAX IpNetToMediaEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

Each entry contains one IpAddress to physical address equivalence.

INDEX { ipNetToMediaIfIndex, ipNetToMediaNetAddress }

::= { ipNetToMediaTable 1 }

IpNetToMediaEntry ::=

SEQUENCE { ipNetToMediaIfIndex INTEGER, ipNetToMediaPhysAddress PhysAddress, ipNetToMediaNetAddress IpAddress, ipNetToMediaType INTEGER }

**ipNetToMediaIfIndex OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

The interface on which this entry's equivalence is effective. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

::= { ipNetToMediaEntry 1 }

#### **ipNetToMediaPhysAddress OBJECT-TYPE**

SYNTAX PhysAddress

ACCESS read-write

STATUS mandatory

DESCRIPTION

The media-dependent physical address.

::= { ipNetToMediaEntry 2 }

#### **ipNetToMediaNetAddress OBJECT-TYPE**

SYNTAX IPAddress

ACCESS read-write

STATUS mandatory

DESCRIPTION

The IPAddress corresponding to the media- dependent physical address.

::= { ipNetToMediaEntry 3 }

#### **ipNetToMediaType OBJECT-TYPE**

SYNTAX Integer { other(1), -- none of the following invalid(2), -- an invalidated mapping  
dynamic(3), static(4) }

ACCESS read-write

STATUS mandatory

DESCRIPTION

The type of mapping. Setting this object to the value invalid(2) has the effect of invalidating the corresponding entry in the ipNetToMediaTable. That is, it effectively disassociates the interface identified with said entry from the mapping identified with said entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant ipNetToMediaType object.

::= { ipNetToMediaEntry 4 }

## 追加の IP オブジェクト

#### **ipRoutingDiscards OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of routing entries which were chosen to be discarded even though they are valid. One possible reason for discarding such an entry could be to free-up buffer space for other routing entries.

::= { ip 23 }

## ICMP グループ

ICMP グループの実装は、すべてのシステムで必須です。

### **icmpInMsgs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of ICMP messages which the entity received. Note that this counter includes all those counted by icmpInErrors.

::= { icmp 1 }

### **icmpInErrors OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP messages which the entity received but determined as having ICMP-specific errors (bad ICMP checksums, bad length, etc.).

::= { icmp 2 }

### **icmpInDestUnreachs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Destination Unreachable messages received.

::= { icmp 3 }

### **icmpInTimeExcds OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Time Exceeded messages received.

::= { icmp 4 }

### **icmpInParmProbs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Parameter Problem messages received.

::= { icmp 5 }

**icmpInSrcQuenches OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Source Quench messages received.

::= { icmp 6 }

**icmpInRedirects OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Redirect messages received.

::= { icmp 7 }

**icmpInEchos OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Echo (request) messages received.

::= { icmp 8 }

**icmpInEchoReps OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Echo Reply messages received.

::= { icmp 9 }

**icmpInTimestamps OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Timestamp (request) messages received.

::= { icmp 10 }

**icmpInTimestampReps OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Timestamp Reply messages received.

::= { icmp 11 }

**icmpInAddrMasks OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Address Mask Request messages received.

::= { icmp 12 }

**icmpInAddrMaskReps OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Address Mask Reply messages received.

::= { icmp 13 }

**icmpOutMsgs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of ICMP messages which this entity attempted to send. Note that this counter includes all those counted by icmpOutErrors.

::= { icmp 14 }

**icmpOutErrors OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP messages which this entity did not send due to problems discovered within ICMP such as a lack of buffers. This value should not include errors discovered outside the ICMP layer such as the inability of IP to route the resultant datagram. In some implementations there may be no types of error which contribute to this counter value.

::= { icmp 15 }

**icmpOutDestUnreachs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Destination Unreachable messages sent.

::= { icmp 16 }

**icmpOutTimeExcds OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Time Exceeded messages sent.

::= { icmp 17 }

**icmpOutParmProbs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Parameter Problem messages sent.

::= { icmp 18 }

**icmpOutSrcQuenchs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Source Quench messages sent.

::= { icmp 19 }

**icmpOutRedirects OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Redirect messages sent. For a host, this object will always be zero, since hosts do not send redirects.

::= { icmp 20 }

**icmpOutEchos OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Echo (request) messages sent.

::= { icmp 21 }

**icmpOutEchoReps OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Echo Reply messages sent.

::= { icmp 22 }

**icmpOutTimestamps OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Timestamp (request) messages sent.

::= { icmp 23 }

**icmpOutTimestampReps OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Timestamp Reply messages sent.

::= { icmp 24 }

**icmpOutAddrMasks OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Address Mask Request messages sent.

::= { icmp 25 }

**icmpOutAddrMaskReps OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of ICMP Address Mask Reply messages sent.

::= { icmp 26 }

## TCP グループ

TCP グループの実装は、TCP を実装するすべてのシステムで必須です。特定の TCP 接続に関する情報を表すオブジェクトタイプのインスタンスは過渡的であり、当該の接続の間しか持続しないことに注意してください。

### tcpRtoAlgorithm OBJECT-TYPE

SYNTAX Integer { other(1), -- none of the following constant(2), -- a constant rto rsre(3), -- MIL-STD-1778, Appendix B vanj(4) -- Van Jacobson's algorithm [10] }

ACCESS read-only

STATUS mandatory

DESCRIPTION

The algorithm used to determine the timeout value used for retransmitting unacknowledged octets.

::= { tcp 1 }

### tcpRtoMin OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

The minimum value permitted by a TCP implementation for the retransmission timeout, measured in milliseconds. More refined semantics for objects of this type depend upon the algorithm used to determine the retransmission timeout. In particular, when the timeout algorithm is rsre(3), an object of this type has the semantics of the LBOUND quantity described in RFC 793.

::= { tcp 2 }

### tcpRtoMax OBJECT-TYPE

SYNTAX Integer

ACCESS read-only

STATUS mandatory

DESCRIPTION

The maximum value permitted by a TCP implementation for the retransmission timeout, measured in milliseconds. More refined semantics for objects of this type depend upon the algorithm used to determine the retransmission timeout. In particular, when the timeout algorithm is rsre(3), an object of this type has the semantics of the UBOUND quantity described in RFC 793.

::= { tcp 3 }

### tcpMaxConn OBJECT-TYPE

SYNTAX Integer



ACCESS read-only

STATUS mandatory

DESCRIPTION

The limit on the total number of TCP connections the entity can support. In entities where the maximum number of connections is dynamic, this object should contain the value -1.

::= { tcp 4 }

#### **tcpActiveOpens OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of times TCP connections have made a direct transition to the SYN-SENT state from the CLOSED state.

::= { tcp 5 }

#### **tcpPassiveOpens OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of times TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state.

::= { tcp 6 }

#### **tcpAttemptFails OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of times TCP connections have made a direct transition to the CLOSED state from either the SYN-SENT state or the SYN-RCVD state, plus the number of times TCP connections have made a direct transition to the LISTEN state from the SYN-RCVD state.

::= { tcp 7 }

#### **tcpEstabResets OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of times TCP connections have made a direct transition to the CLOSED state from either the ESTABLISHED state or the CLOSE-WAIT state.

::= { tcp 8 }

**tcpCurrEstab OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of TCP connections for which the current state is either ESTABLISHED or CLOSE-WAIT.

::= { tcp 9 }

**tcpInSegs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of segments received, including those received in error. This count includes segments received on currently established connections.

::= { tcp 10 }

**tcpOutSegs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of segments sent, including those on current connections but excluding those containing only retransmitted octets.

::= { tcp 11 }

**tcpRetransSegs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of segments retransmitted that is, the number of TCP segments transmitted containing one or more previously transmitted octets.

::= { tcp 12 }

**TCP 接続テーブル**

TCP 接続テーブルには、次の既存のエンティティ TCP 接続に関する情報が格納されています。

**tcpConnTable OBJECT-TYPE**

SYNTAX Sequence of tcpConnEntry

ACCESS not-accessible

STATUS mandatory

## DESCRIPTION

A table containing TCP connection-specific information.

::= { tcp 13 }

**tcpConnEntry OBJECT-TYPE**

SYNTAX TcpConnEntry

ACCESS not-accessible

STATUS mandatory

## DESCRIPTION

Information about a particular current TCP connection. An object of this type is transient, in that it ceases to exist when (or soon after) the connection makes the transition to the CLOSED state.

INDEX { tcpConnLocalAddress, tcpConnLocalPort, tcpConnRemAddress, tcpConnRemPort }

::= { tcpConnTable 1 }

TcpConnEntry ::=

SEQUENCE { tcpConnState INTEGER, tcpConnLocalAddress IpAddress, tcpConnLocalPort INTEGER (0..65535), tcpConnRemAddress IpAddress, tcpConnRemPort INTEGER (0..65535) }

**tcpConnState OBJECT-TYPE**

SYNTAX INTEGER { closed(1), listen(2), synSent(3), synReceived(4), established(5), finWait1(6), finWait2(7), closeWait(8), lastAck(9), closing(10), timeWait(11), deleteTCB(12) }

ACCESS read-write

STATUS mandatory

## DESCRIPTION

The state of this TCP connection. The only value which may be set by a management station is deleteTCB(12). Accordingly, it is appropriate for an agent to return a badValue response if a management station attempts to set this object to any other value. If a management station sets this object to the value deleteTCB(12), then this has the effect of deleting the TCB (as defined in RFC 793) of the corresponding connection on the managed node, resulting in immediate termination of the connection.

As an implementation-specific option, a RST segment may be sent from the managed node to the other TCP endpoint (note however that RST segments are not sent reliably).

::= { tcpConnEntry 1 }

**tcpConnLocalAddress OBJECT-TYPE**

SYNTAX IpAddress

ACCESS read-only

STATUS mandatory

## DESCRIPTION

The local IP address for this TCP connection. In the case of a connection in the listen state which is willing to accept connections for any IP interface associated with the node, the value 0.0.0.0 is used.

::= { tcpConnEntry 2 }

**tcpConnLocalPort OBJECT-TYPE**

SYNTAX INTEGER (0..65535)

ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
 The local port number for this TCP connection.  
 ::= { tcpConnEntry 3 }

**tcpConnRemAddress OBJECT-TYPE**

SYNTAX IpAddress  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
 The remote IP address for this TCP connection.  
 ::= { tcpConnEntry 4 }

**tcpConnRemPort OBJECT-TYPE**

SYNTAX INTEGER (0..65535)  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
 The remote port number for this TCP connection.  
 ::= { tcpConnEntry 5 }

**追加 TCP オブジェクト****tcpInErrs OBJECT-TYPE**

SYNTAX Counter  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
 The total number of segments received in error (e.g., bad TCP checksums).  
 ::= { tcp 14 }

**tcpOutRsts OBJECT-TYPE**

SYNTAX Counter  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
 The number of TCP segments sent containing the RST flag.  
 ::= { tcp 15 }

## UDP グループ

UDP グループの実装は、UDP を実装するすべてのシステムで必須です。

### **udpInDatagrams OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of UDP datagrams delivered to UDP users.

::= { udp 1 }

### **udpNoPorts OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of received UDP datagrams for which there was no application at the destination port.

::= { udp 2 }

### **udpInErrors OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port.

::= { udp 3 }

### **udpOutDatagrams OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of UDP datagrams sent from this entity.

::= { udp 4 }

## UDP リスナー テーブル

UDP リスナー テーブルには、ローカルアプリケーションが現在データグラムを受け入れている、次のエンティティ UDP エンドポイントに関する情報が格納されています。

### **udpTable OBJECT-TYPE**

SYNTAX SEQUENCE OF UdpEntry

ACCESS not-accessible  
 STATUS mandatory  
 DESCRIPTION  
 A table containing UDP listener information.  
 ::= { udp 5 }

**udpEntry OBJECT-TYPE**

SYNTAX UdpEntry  
 ACCESS not-accessible  
 STATUS mandatory  
 DESCRIPTION  
 Information about a particular current UDP listener.  
 INDEX { udpLocalAddress, udpLocalPort }  
 ::= { udpTable 1 }  
 UdpEntry ::=

SEQUENCE { udpLocalAddress IpAddress, udpLocalPort INTEGER (0..65535) }

**udpLocalAddress OBJECT-TYPE**

SYNTAX IpAddress  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
 The local IP address for this UDP listener. In the case of a UDP listener which is willing to accept datagrams for any IP interface associated with the node, the value 0.0.0.0 is used.  
 ::= { udpEntry 1 }

**udpLocalPort OBJECT-TYPE**

SYNTAX INTEGER (0..65535)  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
 The local port number for this UDP listener.  
 ::= { udpEntry 2 }

## EGP グループ

EGP グループの実装は、EGP を実装するすべてのシステムで必須です。

**egpInMsgs OBJECT-TYPE**

SYNTAX Counter  
 ACCESS read-only  
 STATUS mandatory

## DESCRIPTION

The number of EGP messages received without error.

::= { egp 1 }

**egpInErrors OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

## DESCRIPTION

The number of EGP messages received that proved to be in error.

::= { egp 2 }

**egpOutMsgs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

## DESCRIPTION

The total number of locally generated EGP messages.

::= { egp 3 }

**egpOutErrors OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

## DESCRIPTION

The number of locally generated EGP messages not sent due to resource limitations within an EGP entity.

::= { egp 4 }

## EGP ネイバー テーブル

EGP ネイバー テーブルには、次のエンティティ EGP ネイバーに関する情報が格納されています。

**egpNeighTable OBJECT-TYPE**

SYNTAX SEQUENCE OF EgpNeighEntry

ACCESS not-accessible

STATUS mandatory

## DESCRIPTION

The EGP neighbor table.

::= { egp 5 }

**egpNeighEntry OBJECT-TYPE**

SYNTAX EgpNeighEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

Information about this entity's relationship with a particular EGP neighbor.

INDEX { egpNeighAddr }

::= { egpNeighTable 1 }

EgpNeighEntry ::=

SEQUENCE { egpNeighState INTEGER, egpNeighAddr IpAddress, egpNeighAs INTEGER, egpNeighInMsgs Counter, egpNeighInErrs Counter, egpNeighOutMsgs Counter, egpNeighOutErrs Counter, egpNeighInErrMsgs Counter, egpNeighOutErrMsgs Counter, egpNeighStateUps Counter, egpNeighStateDowns Counter, egpNeighIntervalHello INTEGER, egpNeighIntervalPoll INTEGER, egpNeighMode INTEGER, egpNeighEventTrigger INTEGER }

#### **egpNeighState OBJECT-TYPE**

SYNTAX Integer { idle(1), acquisition(2), down(3), up(4), cease(5) }

ACCESS read-only

STATUS mandatory

DESCRIPTION

The EGP state of the local system with respect to the entry EGP neighbor. Each EGP state is represented by a value that is one greater than the numerical value associated with said state in RFC 904.

::= { egpNeighEntry 1 }

#### **egpNeighAddr OBJECT-TYPE**

SYNTAX IpAddress

ACCESS read-only

STATUS mandatory

DESCRIPTION

The IP address of this entry's EGP neighbor.

::= { egpNeighEntry 2 }

#### **egpNeighAs OBJECT-TYPE**

SYNTAX Integer

ACCESS read-only

STATUS mandatory

DESCRIPTION

The autonomous system of this EGP peer. Zero should be specified if the autonomous system number of the neighbor is not yet known.

::= { egpNeighEntry 3 }

#### **egpNeighInMsgs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION



The number of EGP messages received without error from this EGP peer.

::= { egpNeighEntry 4 }

**egpNeighInErrs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of EGP messages received from this EGP peer that proved to be in error (e.g., bad EGP checksum).

::= { egpNeighEntry 5 }

**egpNeighOutMsgs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of locally generated EGP messages to this EGP peer.

::= { egpNeighEntry 6 }

**egpNeighOutErrs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of locally generated EGP messages not sent to this EGP peer due to resource limitations within an EGP entity.

::= { egpNeighEntry 7 }

**egpNeighInErrMsgs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of EGP-defined error messages received from this EGP peer.

::= { egpNeighEntry 8 }

**egpNeighOutErrMsgs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of EGP-defined error messages sent to this EGP peer.

::= { egpNeighEntry 9 }

**egpNeighStateUps OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of EGP state transitions to the UP state with this EGP peer.

::= { egpNeighEntry 10 }

**egpNeighStateDowns OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The number of EGP state transitions from the UP state to any other state with this EGP peer.

::= { egpNeighEntry 11 }

**egpNeighIntervalHello OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

The interval between EGP Hello command retransmissions (in hundredths of a second). This represents the t1 timer as defined in RFC 904.

::= { egpNeighEntry 12 }

**egpNeighIntervalPoll OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

The interval between EGP poll command retransmissions (in hundredths of a second). This represents the t3 timer as defined in RFC 904.

::= { egpNeighEntry 13 }

**egpNeighMode OBJECT-TYPE**

SYNTAX INTEGER { active(1), passive(2) }

ACCESS read-only

STATUS mandatory

DESCRIPTION

The polling mode of this EGP entity, either passive or active.

::= { egpNeighEntry 14 }

**egpNeighEventTrigger OBJECT-TYPE**

SYNTAX INTEGER { start(1), stop(2) }

ACCESS read-write

STATUS mandatory

## DESCRIPTION

A control variable used to trigger operator-initiated Start and Stop events. When read, this variable always returns the most recent value that `egpNeighEventTrigger` was set to. If it has not been set since the last initialization of the network management subsystem on the node, it returns a value of `stop`. When set, this variable causes a Start or Stop event on the specified neighbor, as specified on pages 8-10 of RFC 904. Briefly, a Start event causes an Idle peer to begin neighbor acquisition and a non-Idle peer to reinitiate neighbor acquisition. A stop event causes a non-Idle peer to return to the Idle state until a Start event occurs, either via `egpNeighEventTrigger` or otherwise.

 ::= { `egpNeighEntry` 15 }**追加 EGP オブジェクト****egpAs OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

## DESCRIPTION

The autonomous system number of this EGP entity.

 ::= { `egp` 6 }**トランスミッション グループ**

システム上の各インターフェイスを制御する伝送メディアに基づき、そのシステムには対応するトランスミッショングループが必須となります。伝送メディアを管理するためのインターネット標準が定義されると、トランスミッショングループを使用して、これらのオブジェクト名にプレフィクスを指定します。一般的に、これらの定義は実証されるまで MIB の試行的な部分に属します。その後、インターネット標準化プロセスの一部となり、それに応じてこれらの定義は向上し、トランスミッショングループで新規オブジェクト ID が定義されます。標準では、**type OBJECT IDENTIFIER ::= { transmission number }** という名前が割り当てられます。ここでの **type** は、**ifTable** オブジェクトの **ifType** 列のメディアに使用される記号値であり、**number** はこの記号に対応する実際の整数値です。

**SNMP グループ**

SNMP グループの実装は、SNMP プロトコル エンティティをサポートする、すべてのシステムで必須です。次に定義されるオブジェクトの一部は、管理エージェントまたは管理ステーションのいずれかに特有の機能だけをサポートするために最適化された SNMP 実装において、値がゼロになります。特に、次のオブジェクトは SNMP エンティティを参照することが報告されており、管理対象ノードに複数の SNMP エンティティが常駐する場合があります（たとえば、ノードが管理ステーションとしてホスティングする）。

**snmpInPkts OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of Messages delivered to the SNMP entity from the transport service.

::= { snmp 1 }

**snmpOutPkts OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP Messages which were passed from the SNMP protocol entity to the transport service.

::= { snmp 2 }

**snmpInBadVersions OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP Messages which were delivered to the SNMP protocol entity and were for an unsupported SNMP version.

::= { snmp 3 }

**snmpInBadCommunityNames OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP Messages delivered to the SNMP protocol entity which used a SNMP community name not known to said entity.

::= { snmp 4 }

**snmpInBadCommunityUses OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP Messages delivered to the SNMP protocol entity which represented an SNMP operation which was not allowed by the SNMP community named in the Message.

::= { snmp 5 }

**snmpInASNParseErrs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of ASN.1 or BER errors encountered by the SNMP protocol entity when decoding received SNMP Messages.

::= { snmp 6 }

-- { snmp 7 } is not used

#### **snmpInTooBigs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP PDUs which were delivered to the SNMP protocol entity and for which the value of the error-status field is tooBig.

::= { snmp 8 }

#### **snmpInNoSuchNames OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP PDUs which were delivered to the SNMP protocol entity and for which the value of the error-status field is noSuchName.

::= { snmp 9 }

#### **snmpInBadValues OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP PDUs which were delivered to the SNMP protocol entity and for which the value of the error-status field is `badValue`.

::= { snmp 10 }

#### **snmpInReadOnlys OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number valid SNMP PDUs which were delivered to the SNMP protocol entity and for which the value of the error-status field is readOnly. It should be noted that it is a protocol error to generate an SNMP PDU which contains the value readOnly in the error-status field, as such this object is provided as a means of detecting incorrect implementations of the SNMP.

::= { snmp 11 }

#### **snmpInGenErrs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP PDUs which were delivered to the SNMP protocol entity and for which the value of the error-status field is genErr.

::= { snmp 12 }

#### **snmpInTotalReqVars OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of MIB objects which have been retrieved successfully by the SNMP protocol entity as the result of receiving valid SNMP Get-Request and Get-Next PDUs.

::= { snmp 13 }

#### **snmpInTotalSetVars OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of MIB objects which have been altered successfully by the SNMP protocol entity as the result of receiving valid SNMP Set-Request PDUs.

::= { snmp 14 }

#### **snmpInGetRequests OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP Get-Request PDUs which have been accepted and processed by the SNMP protocol entity.

::= { snmp 15 }

#### **snmpInGetNexts OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP Get-Next PDUs which have been accepted and processed by the SNMP protocol entity.

::= { snmp 16 }

**snmpInSetRequests OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP Set-Request PDUs which have been accepted and processed by the SNMP protocol entity.

::= { snmp 17 }

**snmpInGetResponses OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP Get-Response PDUs which have been accepted and processed by the SNMP protocol entity.

::= { snmp 18 }

**snmpInTraps OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP Trap PDUs which have been accepted and processed by the SNMP protocol entity.

::= { snmp 19 }

**snmpOutTooBig OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP PDUs which were generated by the SNMP protocol entity and for which the value of the error-status field is tooBig.

::= { snmp 20 }

**snmpOutNoSuchNames OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP PDUs which were generated by the SNMP protocol entity and for which the value of the error-status is noSuchName.

::= { snmp 21 }

**snmpOutBadValues OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP PDUs which were generated by the SNMP protocol entity and for which the value of the error-status field is badValue.

::= { snmp 22 }

-- { snmp 23 } is not used

**snmpOutGenErrs OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP PDUs which were generated by the SNMP protocol entity and for which the value of the error-status field is genErr.

::= { snmp 24 }

**snmpOutGetRequests OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP Get-Request PDUs which have been generated by the SNMP protocol entity.

::= { snmp 25 }

**snmpOutGetNexts OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP Get-Next PDUs which have been generated by the SNMP protocol entity.

::= { snmp 26 }

**snmpOutSetRequests OBJECT-TYPE**



SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP Set-Request PDUs which have been generated by the SNMP protocol entity.

::= { snmp 27 }

#### **snmpOutGetResponses OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP Get-Response PDUs which have been generated by the SNMP protocol entity.

::= { snmp 28 }

#### **snmpOutTraps OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

The total number of SNMP Trap PDUs which have been generated by the SNMP protocol entity.

::= { snmp 29 }

#### **snmpEnableAuthenTraps OBJECT-TYPE**

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

ACCESS read-write

STATUS mandatory

DESCRIPTION

Indicates whether the SNMP agent process is permitted to generate authentication-failure traps. The value of this object overrides any configuration information; as such, it provides a means whereby all authentication-failure traps may be disabled. Note that it is strongly recommended that this object be stored in non-volatile memory so that it remains constant between re-initializations of the network management system.

::= { snmp 30 }

## HOST-RESOURCES-MIB



(注)

この HOST-RESOURCE-MIB は形式が変更されています。この項のすべての MIB は、<http://tools.cisco.com/Support/SNMP/do/BrowseMIB.do?local=en&step=2> からダウンロードしてコンパイルしてください。

この MIB では、ホストシステムを管理します。「ホスト」という用語は、インターネットに接続した、類似する他のコンピュータと通信するあらゆるコンピュータであり、1 人以上のユーザによって直接使用されることを意味します。この MIB は、主要な機能として通信サービスを行うデバイス（端末サーバ、ルータ、ブリッジ、モニタリング装置）には必ずしも適用されませんが、このような関連性を明確に妨げません。この MIB に含まれる属性は、たとえば、パーソナルコンピュータや UNIX 変数を実行するシステムといったすべてのインターネット ホストに共通しています。

HOST-RESOURCES-MIB をコンパイルするには、次に示されている順番で MIB をコンパイルする必要があります。

1. SNMPv2-SMI
2. SNMPv2-TC
3. SNMPv2-CONF
4. SNMPv2-MIB
5. IANAifType-MIB
6. IF-MIB
7. RFC1155-SMI
8. RFC-1212
9. SNMPv2-SMI-v1
10. SNMPv2-TC-v1

さらに、次のファイルをダウンロードします。

- OID ファイル : HOST-RESOURCES-MIB.oid

この項の内容は次のとおりです。

- [「改訂」 \(P.8-75\)](#)
- [「定義」 \(P.8-76\)](#)
- [「オブジェクト ID」 \(P.8-76\)](#)
- [「テキストの表記法」 \(P.8-76\)](#)
- [「ホスト リソース システム グループ」 \(P.8-77\)](#)
- [「ホスト リソース ストレージ グループ」 \(P.8-79\)](#)
- [「ホスト リソース デバイス グループ」 \(P.8-81\)](#)
- [「ソフトウェア グループを実行するホスト リソース」 \(P.8-92\)](#)
- [「ソフトウェア パフォーマンス グループを実行するホスト リソース」 \(P.8-95\)](#)
- [「ソフトウェア グループがインストールされたホスト リソース」 \(P.8-96\)](#)
- [「適合情報」 \(P.8-98\)](#)
- [「準拠宣言」 \(P.8-98\)](#)
- [「Cisco Unified CM Release 6.x の機能サービス」 \(P.8-100\)](#)
- [「Cisco Unified CM Release 6.x のネットワーク サービス」 \(P.8-102\)](#)
- [「トラブルシューティング」 \(P.8-103\)](#)

## 改訂

表 8-2 に、最新の改訂から順に MIB の改訂を示します。

表 8-2 改訂の履歴

日付	処置	説明
2000 年 3 月 6 日	追加および更新	<p>実装作業に基づく明確化とバグの修正。この改訂では、SMIv2 形式に変更されています。次の改訂が行われました。</p> <ul style="list-style-type: none"> <li>新規 RFC ドキュメント 標準への形式変更</li> <li>著作権情報の追加</li> <li>SNMP フレームワークの概要の更新</li> <li>参照項目の更新</li> <li>RFC 2119 への参照を追加</li> <li>重要なセキュリティ考慮事項の追加</li> </ul> <p>新規タイプの登録に関する IANA の新しい考慮事項、次のタイプおよびマクロに関する新規 SMIv2 構文への変換</p> <ul style="list-style-type: none"> <li>Counter32、Integer32、Gauge32、MODULE-IDENTITY、OBJECT-TYPE、TEXTUAL-CONVENTION、OBJECT-IDENTITY、MODULE-COMPLIANCE、OBJECT-GROUP</li> <li>新しいテキストの表記法を使用：TruthValue、DateAndTime、AutonomousType、InterfaceIndexOrZero</li> <li>hrPrinterStatus の誤字の修正</li> <li>hrPrinterDetectedErrorState に不足していたエラー ビットを追加</li> <li>hrPrinterStatus への推奨マッピングに起因する不明点を明確化しました。</li> <li>InternationalDisplayString のオブジェクトタイプのサイズを明確化し、符号化した記号ではなく、オクテット数にしました。</li> <li>実装作業に基づき、次のオブジェクトの使用を明確化しました。 hrSystemInitialLoadDevice、hrSystemInitialLoadParameters、hrMemorySize、hrStorageSize、hrStorageAllocationFailures、hrDeviceErrors、hrProcessorLoad、hrNetworkIfIndex、hrDiskStorageCapacity、hrSWRunStatus、hrSWRunPerfCPU、および hrSWInstalledDate</li> <li>hrSWInstalledTable に関する実装手法を明確化しました。</li> <li>hrSWRunPerfTable に新しい AUGMENT 句を使用しています。</li> <li>国際化に関する考慮事項を追加しました。この改訂は RFC2790 として公表されています。</li> </ul>
1999 年 10 月 20 日	初版	<p>この MIB のオリジナルバージョンは、RFC1514 として公表されています。 ::= { hrMIBAdminInfo 1 }</p>

## 定義

次の定義が HOST-RESOURCES-MIB 用にインポートされています。

- MODULE-IDENTITY、OBJECT-TYPE、mib-2、Integer32、Counter32、Gauge32、TimeTicks
- SNMPv2-SMI から : TEXTUAL-CONVENTION、DisplayString、TruthValue、DateAndTime、AutonomousType
- SNMPv2-TC から : MODULE-COMPLIANCE、OBJECT-GROUP
- SNMPv2-CONF から : InterfaceIndexOrZero
- IF-MIB から : hostResourcesMibModule MODULE-IDENTITY

## オブジェクト ID

```

host OBJECT IDENTIFIER ::= { mib-2 25 }
hrSystem OBJECT IDENTIFIER ::= { host 1 }
hrStorage OBJECT IDENTIFIER ::= { host 2 }
hrDevice OBJECT IDENTIFIER ::= { host 3 }
hrSWRun OBJECT IDENTIFIER ::= { host 4 }
hrSWRunPerf OBJECT IDENTIFIER ::= { host 5 }
hrSWInstalled OBJECT IDENTIFIER ::= { host 6 }
hrMIBAdminInfo OBJECT IDENTIFIER ::= { host 7 }

```

## テキストの表記法

### **KBytes ::= TEXTUAL-CONVENTION**

```

STATUS current
DESCRIPTION
Storage size, expressed in units of 1024 bytes.
SYNTAX Integer32 (0..2147483647)

```

### **ProductID ::= TEXTUAL-CONVENTION**

```

STATUS current
DESCRIPTION
This textual convention is intended to identify the manufacturer, model, and version of a specific hardware or software product. It is suggested that these OBJECT IDENTIFIERS are allocated such that all products from a particular manufacturer are registered under a subtree distinct to that manufacturer. In addition, all versions of a product should be registered under a subtree distinct to that product. With this strategy, a management station may uniquely determine the manufacturer and/or model of a product whose productID is unknown to the management station. Objects of this type may be useful for inventory purposes or for automatically detecting incompatibilities or version mismatches between various hardware and software components on a system.

```

For example, the product ID for the ACME 4860 66MHz clock doubled processor might be: enterprises.acme.acmeProcessors.a4860DX2.MHz66.A software product might be registered as: enterprises.acme.acmeOperatingSystems.acmeDOS.six(6).one(1).

SYNTAX OBJECT IDENTIFIER

UnknownProduct will be used for any unknown ProductID.UnknownProduct OBJECT IDENTIFIER ::= { 0 0 }

**InternationalDisplayString ::= TEXTUAL-CONVENTION**

STATUS current

DESCRIPTION

This data type is used to model textual information in some character set. A network management station should use a local algorithm to determine which character set is in use and how it should be displayed. Note that this character set may be encoded with more than one octet per symbol, but will most often be NVT ASCII. When a size clause is specified for an object of this type, the size refers to the length in octets, not the number of symbols.

SYNTAX OCTET STRING

## ホスト リソース システム グループ

**hrSystemUptime OBJECT-TYPE**

SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The amount of time since this host was last initialized. Note that this is different from sysUpTime in the SNMPv2-MIB [RFC1907] because sysUpTime is the uptime of the network management portion of the system.

::= { hrSystem 1 }

**hrSystemDate OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-write

STATUS current

DESCRIPTION

The host's notion of the local date and time of day.

::= { hrSystem 2 }

**hrSystemInitialLoadDevice OBJECT-TYPE**

SYNTAX Integer32 (1..2147483647)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

The index of the hrDeviceEntry for the device from which this host is configured to load its initial operating system configuration (i.e., which operating system code and/or boot parameters). Note that writing to this object just changes the configuration that will be used the next time the operating system is loaded and does not actually cause the reload to occur.

::= { hrSystem 3 }

#### **hrSystemInitialLoadParameters OBJECT-TYPE**

SYNTAX InternationalDisplayString (SIZE (0..128))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

This object contains the parameters (e.g. a pathname and parameter) supplied to the load device when requesting the initial operating system configuration from that device. Note that writing to this object just changes the configuration that will be used the next time the operating system is loaded and does not actually cause the reload to occur.

::= { hrSystem 4 }

#### **hrSystemNumUsers OBJECT-TYPE**

SYNTAX Gauge32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of user sessions for which this host is storing state information. A session is a collection of processes requiring a single act of user authentication and possibly subject to collective job control.

::= { hrSystem 5 }

#### **hrSystemProcesses OBJECT-TYPE**

SYNTAX Gauge32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of process contexts currently loaded or running on this system.

::= { hrSystem 6 }

#### **hrSystemMaxProcesses OBJECT-TYPE**

SYNTAX Integer32 (0..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The maximum number of process contexts this system can support. If there is no fixed maximum, the value should be zero. On systems that have a fixed maximum, this object can help diagnose failures that occur when this maximum is reached.

::= { hrSystem 7 }

## ホスト リソース ストレージ グループ

hrStorageType と併用する場合、ストレージタイプの登録ポイントです。これらは、HOST-RESOURCES-TYPES モジュールで定義されています。

hrStorageTypes OBJECT IDENTIFIER ::= { hrStorage 1 }

hrMemorySize OBJECT-TYPE

SYNTAX KBytes

UNITS KBytes

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The amount of physical read-write main memory, typically RAM, contained by the host.

::= { hrStorage 2 }

### hrStorageTable OBJECT-TYPE

SYNTAX Sequence of HrStorageEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The (conceptual) table of logical storage areas on the host. An entry shall be placed in the storage table for each logical area of storage that is allocated and has fixed resource limits. The amount of storage represented in an entity is the amount actually usable by the requesting entity, and excludes loss due to formatting or file system reference information.

These entries are associated with logical storage areas, as might be seen by an application, rather than physical storage entities which are typically seen by an operating system. Storage such as tapes and floppies without file systems on them are typically not allocated in chunks by the operating system to requesting applications, and therefore shouldn't appear in this table. Examples of valid storage for this table include disk partitions, file systems, RAM (for some architectures this is further segmented into regular memory, extended memory, and so on), backing store for virtual memory ('swap space').

This table is intended to be a useful diagnostic for "out of memory" and "out of buffers" types of failures. In addition, it can be a useful performance monitoring tool for tracking memory, disk, or buffer usage.

::= { hrStorage 3 }

### hrStorageEntry OBJECT-TYPE

SYNTAX HrStorageEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

A (conceptual) entry for one logical storage area on the host. As an example, an instance of the hrStorageType object might be named hrStorageType.3

INDEX { hrStorageIndex }

::= { hrStorageTable 1 }

hrStorageEntry ::= SEQUENCE { hrStorageIndex Integer32, hrStorageTypeAutonomousType, hrStorageDescr DisplayString, hrStorageAllocationUnits Integer32, hrStorageSizeInteger32, hrStorageUsedInteger32, hrStorageAllocationFailures Counter32 }

**hrStorageIndex OBJECT-TYPE**

SYNTAX Integer32 (1..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A unique value for each logical storage area contained by the host.

::= { hrStorageEntry 1 }

**hrStorageType OBJECT-TYPE**

SYNTAX AutonomousType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The type of storage represented by this entry.

::= { hrStorageEntry 2 }

**hrStorageDescr OBJECT-TYPE**

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A description of the type and instance of the storage described by this entry.

::= { hrStorageEntry 3 }

**hrStorageAllocationUnits OBJECT-TYPE**

SYNTAX Integer32 (1..2147483647)

UNITS Bytes

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The size, in bytes, of the data objects allocated from this pool. If this entry is monitoring sectors, blocks, buffers, or packets, for example, this number will commonly be greater than one. Otherwise this number will typically be one.

::= { hrStorageEntry 4 }

**hrStorageSize OBJECT-TYPE**

SYNTAX Integer32 (0..2147483647)

MAX-ACCESS read-write

STATUS current

DESCRIPTION



The size of the storage represented by this entry, in units of `hrStorageAllocationUnits`. This object is writable to allow remote configuration of the size of the storage area in those cases where such an operation makes sense and is possible on the underlying system. For example, the amount of main memory allocated to a buffer pool might be modified or the amount of disk space allocated to virtual memory might be modified.

::= { hrStorageEntry 5 }

#### **hrStorageUsed OBJECT-TYPE**

SYNTAX Integer32 (0..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The amount of the storage represented by this entry that is allocated, in units of `hrStorageAllocationUnits`.

::= { hrStorageEntry 6 }

#### **hrStorageAllocationFailures OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of requests for storage represented by this entry that could not be honored due to not enough storage. It should be noted that as this object has a SYNTAX of Counter32, that it does not have a defined initial value. However, it is recommended that this object be initialized to zero, even though management stations must not depend on such an initialization.

::= { hrStorageEntry 7 }

## ホスト リソース デバイス グループ

このデバイス グループは、システム上のデバイスの識別と診断に役立ちます。 `hrDeviceTable` には、あらゆるタイプのデバイスに共通の情報が格納されています。さらに、デバイスによっては、詳細情報のためのデバイス特有のテーブルを保有しています。将来的に、その他のデバイス タイプにも、こうしたテーブルが定義される可能性があります。 `hrDeviceType` と併用する場合、デバイス タイプの登録ポインタです。これらは、HOST-RESOURCES-TYPES モジュールで定義されています。

hrDeviceTypes OBJECT IDENTIFIER ::= { hrDevice 1 }

#### **hrDeviceTable OBJECT-TYPE**

SYNTAX Sequence of hrDeviceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The (conceptual) table of devices contained by the host.

::= { hrDevice 2 }

#### **hrDeviceEntry OBJECT-TYPE**

SYNTAX hrDeviceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

A (conceptual) entry for one device contained by the host. As an example, an instance of the hrDeviceType object might be named hrDeviceType.3

INDEX { hrDeviceIndex }

::= { hrDeviceTable 1 }

HrDeviceEntry ::= SEQUENCE { hrDeviceIndex Integer32, hrDeviceTypeAutonomousType, hrDeviceDescr DisplayString, hrDeviceID ProductID, hrDeviceStatus INTEGER, hrDeviceErrors Counter32 }

#### **hrDeviceIndex OBJECT-TYPE**

SYNTAX Integer32 (1..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A unique value for each device contained by the host. The value for each device must remain constant at least from one re-initialization of the agent to the next re-initialization.

::= { hrDeviceEntry 1 }

#### **hrDeviceType OBJECT-TYPE**

SYNTAX AutonomousType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

An indication of the type of device. If this value is “hrDeviceProcessor { hrDeviceTypes 3 }” then an entry exists in the hrProcessorTable which corresponds to this device. If this value is “hrDeviceNetwork { hrDeviceTypes 4 }”, then an entry exists in the hrNetworkTable which corresponds to this device. If this value is “hrDevicePrinter { hrDeviceTypes 5 }”, then an entry exists in the hrPrinterTable which corresponds to this device.

If this value is “hrDeviceDiskStorage { hrDeviceTypes 6 }”, then an entry exists in the hrDiskStorageTable which corresponds to this device.

::= { hrDeviceEntry 2 }

#### **hrDeviceDescr OBJECT-TYPE**

SYNTAX DisplayString (SIZE (0..64))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A textual description of this device, including the device's manufacturer and revision, and optionally, its serial number.

::= { hrDeviceEntry 3 }

**hrDeviceID OBJECT-TYPE**

SYNTAX ProductID

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The product ID for this device.

::= { hrDeviceEntry 4 }

**hrDeviceStatus OBJECT-TYPE**

SYNTAX INTEGER { unknown(1), running(2), warning(3), testing(4), down(5) }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The current operational state of the device described by this row of the table. A value unknown(1) indicates that the current state of the device is unknown. running(2) indicates that the device is up and running and that no unusual error conditions are known. The warning(3) state indicates that agent has been informed of an unusual error condition by the operational software (e.g., a disk device driver) but that the device is still 'operational'. An example would be a high number of soft errors on a disk. A value of testing(4), indicates that the device is not available for use because it is in the testing state. The state of down(5) is used only when the agent has been informed that the device is not available for any use.

::= { hrDeviceEntry 5 }

**hrDeviceErrors OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of errors detected on this device. It should be noted that as this object has a SYNTAX of Counter32, that it does not have a defined initial value. However, it is recommended that this object be initialized to zero, even though management stations must not depend on such an initialization.

::= { hrDeviceEntry 6 }

**hrProcessorTable OBJECT-TYPE**

SYNTAX Sequence of hrProcessorEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The (conceptual) table of processors contained by the host. Note that this table is potentially sparse: a (conceptual) entry exists only if the correspondent value of the hrDeviceType object is hrDeviceProcessor.

::= { hrDevice 3 }

**hrProcessorEntry OBJECT-TYPE**

SYNTAX hrProcessorEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

A (conceptual) entry for one processor contained by the host. The hrDeviceIndex in the index represents the entry in the hrDeviceTable that corresponds to the hrProcessorEntry. As an example of how objects in this table are named, an instance of the hrProcessorFrwID object might be named hrProcessorFrwID.3

INDEX { hrDeviceIndex }

::= { hrProcessorTable 1 }

HrProcessorEntry ::= SEQUENCE { hrProcessorFrwIDProductID, hrProcessorLoad Integer32 }

#### **hrProcessorFrwID OBJECT-TYPE**

SYNTAX ProductID

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The product ID of the firmware associated with the processor.

::= { hrProcessorEntry 1 }

#### **hrProcessorLoad OBJECT-TYPE**

SYNTAX Integer32 (0..100)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The average, over the last minute, of the percentage of time that this processor was not idle. Implementations may approximate this one minute smoothing period if necessary.

::= { hrProcessorEntry 2 }

#### **hrNetworkTable OBJECT-TYPE**

SYNTAX Sequence of hrNetworkEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The (conceptual) table of network devices contained by the host. Note that this table is potentially sparse: a (conceptual) entry exists only if the correspondent value of the hrDeviceType object is hrDeviceNetwork.

::= { hrDevice 4 }

#### **hrNetworkEntry OBJECT-TYPE**

SYNTAX hrNetworkEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

A (conceptual) entry for one network device contained by the host. The hrDeviceIndex in the index represents the entry in the hrDeviceTable that corresponds to the hrNetworkEntry. As an example of how objects in this table are named, an instance of the hrNetworkIfIndex object might be named hrNetworkIfIndex.3.

INDEX { hrDeviceIndex }

::= { hrNetworkTable 1 }

hrNetworkEntry ::= SEQUENCE { hrNetworkIfIndexInterfaceIndexOrZero }

#### **hrNetworkIfIndex OBJECT-TYPE**

SYNTAX InterfaceIndexOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The value of ifIndex which corresponds to this network device. If this device is not represented in the ifTable, then this value shall be zero.

::= { hrNetworkEntry 1 }

#### **hrPrinterTable OBJECT-TYPE**

SYNTAX Sequence of hrPrinterEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The (conceptual) table of printers local to the host. Note that this table is potentially sparse: a (conceptual) entry exists only if the correspondent value of the hrDeviceType object is hrDevicePrinter.

::= { hrDevice 5 }

#### **hrPrinterEntry OBJECT-TYPE**

SYNTAX hrPrinterEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

A (conceptual) entry for one printer local to the host. The hrDeviceIndex in the index represents the entry in the hrDeviceTable that corresponds to the hrPrinterEntry.

As an example of how objects in this table are named, an instance of the hrPrinterStatus object might be named hrPrinterStatus.3

INDEX { hrDeviceIndex }

::= { hrPrinterTable 1 }

hrPrinterEntry ::= SEQUENCE { hrPrinterStatus INTEGER, hrPrinterDetectedErrorState OCTET STRING }

#### **hrPrinterStatus OBJECT-TYPE**

SYNTAX INTEGER { other(1), unknown(2), idle(3), printing(4), warmup(5) }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The current status of this printer device.

::= { hrPrinterEntry 1 }

#### **hrPrinterDetectedErrorState OBJECT-TYPE**

SYNTAX OCTET STRING

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object represents any error conditions detected by the printer. The error conditions are encoded as bits in an octet string, with the following definitions (condition first then bit number):

- lowPaper 0
- noPaper 1
- lowToner 2
- noToner 3
- doorOpen 4
- jammed5
- offline 6
- serviceRequested 7
- inputTrayMissing 8
- outputTrayMissing 9
- markerSupplyMissing 10
- outputNearFull 11
- outputFull 12
- inputTrayEmpty 13
- overduePreventMaint 14

Bits are numbered starting with the most significant bit of the first byte being bit 0, the least significant bit of the first byte being bit 7, the most significant bit of the second byte being bit 8, and so on. A one bit encodes that the condition was detected, while a zero bit encodes that the condition was not detected.

This object is useful for alerting an operator to specific warning or error conditions that may occur, especially those requiring human intervention.

::= { hrPrinterEntry 2 }

#### **hrDiskStorageTable OBJECT-TYPE**

SYNTAX Sequence of hrDiskStorageEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The (conceptual) table of long-term storage devices contained by the host. In particular, disk devices accessed remotely over a network are not included here. Note that this table is potentially sparse: a (conceptual) entry exists only if the correspondent value of the hrDeviceType object is hrDeviceDiskStorage.

::= { hrDevice 6 }

#### **hrDiskStorageEntry OBJECT-TYPE**

SYNTAX hrDiskStorageEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

A (conceptual) entry for one long-term storage device contained by the host. The hrDeviceIndex in the index represents the entry in the hrDeviceTable that corresponds to the hrDiskStorageEntry. As an example, an instance of the hrDiskStorageCapacity object might be named hrDiskStorageCapacity.3

INDEX { hrDeviceIndex }

::= { hrDiskStorageTable 1 }

hrDiskStorageEntry ::= SEQUENCE { hrDiskStorageAccess INTEGER, hrDiskStorageMedia INTEGER, hrDiskStorageRemoveable TruthValue, hrDiskStorageCapacity KBytes }

#### **hrDiskStorageAccess OBJECT-TYPE**

SYNTAX INTEGER { readWrite(1), readOnly(2) }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

An indication if this long-term storage device is readable and writable or only readable. This should reflect the media type, any write-protect mechanism, and any device configuration that affects the entire device.

::= { hrDiskStorageEntry 1 }

#### **hrDiskStorageMedia OBJECT-TYPE**

SYNTAX INTEGER { other(1), unknown(2), hardDisk(3), floppyDisk(4), opticalDiskROM(5), opticalDiskWORM(6), --Write Once Read Many-- opticalDiskRW(7), ramDisk(8) }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

An indication of the type of media used in this long-term storage device.

::= { hrDiskStorageEntry 2 }

#### **hrDiskStorageRemoveable OBJECT-TYPE**

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

DESCRIPTION

Denotes whether or not the disk media may be removed from the drive.

::= { hrDiskStorageEntry 3 }

#### **hrDiskStorageCapacity OBJECT-TYPE**

SYNTAX KBytes

UNITS KBytes

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The total size for this long-term storage device.If the media is removable and is currently removed, this value should be zero.

::= { hrDiskStorageEntry 4 }

#### **hrPartitionTable OBJECT-TYPE**

SYNTAX Sequence of hrPartitionEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The (conceptual) table of partitions for long-term storage devices contained by the host.In particular, partitions accessed remotely over a network are not included here.

::= { hrDevice 7 }

#### **hrPartitionEntry OBJECT-TYPE**

SYNTAX hrPartitionEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

A (conceptual) entry for one partition.The hrDeviceIndex in the index represents the entry in the hrDeviceTable that corresponds to the hrPartitionEntry.

As an example of how objects in this table are named, an instance of the hrPartitionSize object might be named hrPartitionSize.3.1

INDEX { hrDeviceIndex, hrPartitionIndex }

::= { hrPartitionTable 1 }

hrPartitionEntry ::= SEQUENCE { hrPartitionIndexInteger32,  
hrPartitionLabelInternationalDisplayString, hrPartitionID OCTET STRING, hrPartitionSize Bytes,  
hrPartitionFSIndex Integer32 }

#### **hrPartitionIndex OBJECT-TYPE**

SYNTAX Integer32 (1..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION



A unique value for each partition on this long-term storage device. The value for each long-term storage device must remain constant at least from one re-initialization of the agent to the next re-initialization.

::= { hrPartitionEntry 1 }

**hrPartitionLabel OBJECT-TYPE**

SYNTAX InternationalDisplayString (SIZE (0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A textual description of this partition.

::= { hrPartitionEntry 2 }

**hrPartitionID OBJECT-TYPE**

SYNTAX OCTET STRING

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A descriptor which uniquely represents this partition to the responsible operating system. On some systems, this might take on a binary representation.

::= { hrPartitionEntry 3 }

**hrPartitionSize OBJECT-TYPE**

SYNTAX KBytes

UNITS KBytes

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The size of this partition.

::= { hrPartitionEntry 4 }

**hrPartitionFSIndex OBJECT-TYPE**

SYNTAX Integer32 (0..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The index of the file system mounted on this partition. If no file system is mounted on this partition, then this value shall be zero. Note that multiple partitions may point to one file system, denoting that that file system resides on those partitions. Multiple file systems may not reside on one partition.

::= { hrPartitionEntry 5 }

## ファイル システム テーブル

hrFSType で使用する、一般的なファイル システム タイプの登録ポイントです。これらは、HOST-RESOURCES-TYPES モジュールで定義されています。

**hrFSTypes OBJECT IDENTIFIER ::= { hrDevice 9 }**

### hrFSTable OBJECT-TYPE

SYNTAX Sequence of hrFSEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The (conceptual) table of file systems local to this host or remotely mounted from a file server. File systems that are in only one user's environment on a multi-user system will not be included in this table.

::= { hrDevice 8 }

### hrFSEntry OBJECT-TYPE

SYNTAX hrFSEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

A (conceptual) entry for one file system local to this host or remotely mounted from a file server. File systems that are in only one user's environment on a multi-user system will not be included in this table.

As an example of how objects in this table are named, an instance of the hrFSMountPoint object might be named hrFSMountPoint.3

INDEX { hrFSIndex }

::= { hrFSTable 1 }

hrFSEntry ::= SEQUENCE { hrFSIndex Integer32, hrFSMountPoint InternationalDisplayString, hrFSRemoteMountPointInternationalDisplayString, hrFSTypeAutonomousType, hrFSAccess INTEGER, hrFSBootableTruthValue, hrFSStorageIndexInteger32, hrFSLastFullBackupDate DateAndTime, hrFSLastPartialBackupDate DateAndTime }

### hrFSIndex OBJECT-TYPE

SYNTAX Integer32 (1..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A unique value for each file system local to this host. The value for each file system must remain constant at least from one re-initialization of the agent to the next re-initialization.

::= { hrFSEntry 1 }

### hrFSMountPoint OBJECT-TYPE

SYNTAX InternationalDisplayString (SIZE(0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The path name of the root of this file system.

::= { hrFSEntry 2 }

#### **hrFSRemoteMountPoint OBJECT-TYPE**

SYNTAX InternationalDisplayString (SIZE(0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A description of the name and/or address of the server that this file system is mounted from. This may also include parameters such as the mount point on the remote file system. If this is not a remote file system, this string should have a length of zero.

::= { hrFSEntry 3 }

#### **hrFSType OBJECT-TYPE**

SYNTAX AutonomousType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The value of this object identifies the type of this file system.

::= { hrFSEntry 4 }

#### **hrFSAccess OBJECT-TYPE**

SYNTAX Integer { readWrite(1), readOnly(2) }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

An indication if this file system is logically configured by the operating system to be readable and writable or only readable. This does not represent any local access-control policy, except one that is applied to the file system as a whole.

::= { hrFSEntry 5 }

#### **hrFSBootable OBJECT-TYPE**

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A flag indicating whether this file system is bootable.

::= { hrFSEntry 6 }

#### **hrFSStorageIndex OBJECT-TYPE**

SYNTAX Integer32 (0..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The index of the hrStorageEntry that represents information about this file system. If there is no such information available, then this value shall be zero. The relevant storage entry will be useful in tracking the percent usage of this file system and diagnosing errors that may occur when it runs out of space.

::= { hrFSEntry 7 }

#### hrFSLastFullBackupDate OBJECT-TYPE

SYNTAX DateAndTime

MAX-ACCESS read-write

STATUS current

DESCRIPTION

The last date at which this complete file system was copied to another storage device for backup. This information is useful for ensuring that backups are being performed regularly. If this information is not known, then this variable shall have the value corresponding to January 1, year 0000, 00:00:00.0, which is encoded as (hex) 00 00 01 01 00 00 00 00.

::= { hrFSEntry 8 }

#### hrFSLastPartialBackupDate OBJECT-TYPE

SYNTAX DateAndTime

MAX-ACCESS read-write

STATUS current

DESCRIPTION

The last date at which a portion of this file system was copied to another storage device for backup. This information is useful for ensuring that backups are being performed regularly. If this information is not known, then this variable shall have the value corresponding to January 1, year 0000, 00:00:00.0, which is encoded as (hex) 00 00 01 01 00 00 00 00.

::= { hrFSEntry 9 }

## ソフトウェア グループを実行するホスト リソース

hrSWRunTable には、実行中であるか、実行に備えて物理メモリまたは仮想メモリにロードされている各ソフトウェアのエントリが格納されています。これには、ホストのオペレーティング システム、デバイス ドライバ、アプリケーションなどがあります。

#### hrSWOSIndex OBJECT-TYPE

SYNTAX Integer32 (1..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The value of the hrSWRunIndex for the hrSWRunEntry that represents the primary operating system running on this host. This object is useful for quickly and uniquely identifying that primary operating system.

::= { hrSWRun 1 }

**hrSWRunTable OBJECT-TYPE**

SYNTAX Sequence of hrSWRunEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The (conceptual) table of software running on the host.

::= { hrSWRun 2 }

**hrSWRunEntry OBJECT-TYPE**

SYNTAX hrSWRunEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

A (conceptual) entry for one piece of software running on the host Note that because the installed software table only contains information for software stored locally on this host, not every piece of running software will be found in the installed software table. This is true of software that was loaded and run from a non-local source, such as a network-mounted file system.

As an example of how objects in this table are named, an instance of the hrSWRunName object might be named hrSWRunName.1287

INDEX { hrSWRunIndex }

::= { hrSWRunTable 1 }

```
HrSWRunEntry ::= SEQUENCE {
    hrSWRunIndex Integer32,
    hrSWRunNameInternationalDisplayString,
    hrSWRunID ProductID,
    hrSWRunPathInternationalDisplayString,
    hrSWRunParameters InternationalDisplayString,
    hrSWRunTypeINTEGER,
    hrSWRunStatus INTEGER }
```

**hrSWRunIndex OBJECT-TYPE**

SYNTAX Integer32 (1..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A unique value for each piece of software running on the host. Wherever possible, this should be the system's native, unique identification number.

::= { hrSWRunEntry 1 }

**hrSWRunName OBJECT-TYPE**

SYNTAX InternationalDisplayString (SIZE (0..64))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A textual description of this running piece of software, including the manufacturer, revision, and the name by which it is commonly known. If this software was installed locally, this should be the same string as used in the corresponding hrSWInstalledName.

::= { hrSWRunEntry 2 }

**hrSWRunID OBJECT-TYPE**

SYNTAX ProductID

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The product ID of this running piece of software.

::= { hrSWRunEntry 3 }

**hrSWRunPath OBJECT-TYPE**

SYNTAX InternationalDisplayString (SIZE(0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A description of the location on long-term storage (e.g. a disk drive) from which this software was loaded.

::= { hrSWRunEntry 4 }

**hrSWRunParameters OBJECT-TYPE**

SYNTAX InternationalDisplayString (SIZE(0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A description of the parameters supplied to this software when it was initially loaded.

::= { hrSWRunEntry 5 }

**hrSWRunType OBJECT-TYPE**

SYNTAX INTEGER { unknown(1), operatingSystem(2), deviceDriver(3), application(4) }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The type of this software.

::= { hrSWRunEntry 6 }

**hrSWRunStatus OBJECT-TYPE**

SYNTAX INTEGER { running(1), runnable(2), -- waiting for resource -- (i.e., CPU, memory, IO) notRunnable(3), -- loaded but waiting for event invalid(4) -- not loaded }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

The status of this running piece of software. Setting this value to invalid(4) shall cause this software to stop running and to be unloaded. Sets to other values are not valid.

```
::= { hrSWRunEntry 7 }
```

## ソフトウェア パフォーマンス グループを実行するホスト リソース

hrSWRunPerfTable には、hrSWRunTable の各エントリに対応するエントリが格納されています。

### hrSWRunPerfTable OBJECT-TYPE

SYNTAX Sequence of hrSWRunPerfEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The (conceptual) table of running software performance metrics.

```
::= { hrSWRunPerf 1 }
```

### hrSWRunPerfEntry OBJECT-TYPE

SYNTAX hrSWRunPerfEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

A (conceptual) entry containing software performance metrics. As an example, an instance of the hrSWRunPerfCPU object might be named hrSWRunPerfCPU.1287. This table augments information in the hrSWRunTable.

AUGMENTS { hrSWRunEntry }

```
::= { hrSWRunPerfTable 1 }
```

```
hrSWRunPerfEntry ::= SEQUENCE { hrSWRunPerfCPU Integer32, hrSWRunPerfMem KBytes }
```

### hrSWRunPerfCPU OBJECT-TYPE

SYNTAX Integer32 (0..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of centi-seconds of the total system's CPU resources consumed by this process. Note that on a multi-processor system, this value may increment by more than one centi-second in one centi-second of real (wall clock) time.

```
::= { hrSWRunPerfEntry 1 }
```

### hrSWRunPerfMem OBJECT-TYPE

SYNTAX KBytes

UNITS KBytes

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The total amount of real system memory allocated to this process.

```
::= { hrSWRunPerfEntry 2 }
```

## ソフトウェア グループがインストールされたホスト リソース

hrSWInstalledTable には、このホストの長期ストレージ（ディスク ドライブなど）にローカルにインストールされた各ソフトウェアのエントリが格納されています。これにはネットワーク サーバからリモートでロードできるソフトウェアは含まれないので注意してください。異なる実装では、さまざまな方法でソフトウェアをトラッキングすることがあります。たとえば、ある実装では別のソフトウェアとして実行可能ファイルをトラッキングし、別の実装ではソフトウェア パッケージ（関連するファイルのグループなど）のトラッキングや、システムまたはアプリケーション パッチのトラッキングを行うなどの方法が使用されます。

このテーブルは、ホスト上のソフトウェアを識別してそのコンポーネントを作成したり、さまざまなハードウェアおよびソフトウェア間の非互換性やバージョンの不一致の問題を診断したりする場合に役立ちます。

### hrSWInstalledLastChange OBJECT-TYPE

SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The value of sysUpTime when an entry in the hrSWInstalledTable was last added, renamed, or deleted. Because this table is likely to contain many entries, polling of this object allows a management station to determine when re-downloading of the table might be useful.

```
::= { hrSWInstalled 1 }
```

### hrSWInstalledLastUpdateTime OBJECT-TYPE

SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The value of sysUpTime when the hrSWInstalledTable was last completely updated. Because caching of this data will be a popular implementation strategy, retrieval of this object allows a management station to obtain a guarantee that no data in this table is older than the indicated time.

```
::= { hrSWInstalled 2 }
```

### hrSWInstalledTable OBJECT-TYPE

SYNTAX SEQUENCE OF HrSWInstalledEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The (conceptual) table of software installed on this host.

```
::= { hrSWInstalled 3 }
```

### hrSWInstalledEntry OBJECT-TYPE

SYNTAX HrSWInstalledEntry



MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

A (conceptual) entry for a piece of software installed on this host. As an example of how objects in this table are named, an instance of the hrSWInstalledName object might be named hrSWInstalledName.96

INDEX { hrSWInstalledIndex }

::= { hrSWInstalledTable 1 }

hrSWInstalledEntry ::= SEQUENCE { hrSWInstalledIndex Integer32, hrSWInstalledNameInternationalDisplayString, hrSWInstalledID ProductID, hrSWInstalledTypeINTEGER, hrSWInstalledDateDateAndTime }

#### **hrSWInstalledIndex OBJECT-TYPE**

SYNTAX Integer32 (1..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A unique value for each piece of software installed on the host. This value shall be in the range from 1 to the number of pieces of software installed on the host.

::= { hrSWInstalledEntry 1 }

#### **hrSWInstalledName OBJECT-TYPE**

SYNTAX InternationalDisplayString (SIZE (0..64))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A textual description of this installed piece of software, including the manufacturer, revision, the name by which it is commonly known, and optionally, its serial number.

::= { hrSWInstalledEntry 2 }

#### **hrSWInstalledID OBJECT-TYPE**

SYNTAX ProductID

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The product ID of this installed piece of software.

::= { hrSWInstalledEntry 3 }

#### **hrSWInstalledType OBJECT-TYPE**

SYNTAX INTEGER { unknown(1), operatingSystem(2), deviceDriver(3), application(4) }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The type of this software.

::= { hrSWInstalledEntry 4 }

#### **hrSWInstalledDate OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The last-modification date of this application as it would appear in a directory listing.

If this information is not known, then this variable shall have the value corresponding to January 1, year 0000, 00:00:00.0, which is encoded as (hex) 00 00 01 01 00 00 00 00.

::= { hrSWInstalledEntry 5 }

## 適合情報

hrMIBCompliances OBJECT IDENTIFIER ::= { hrMIBAdminInfo 2 }

hrMIBGroups OBJECT IDENTIFIER ::= { hrMIBAdminInfo 3 }

## 準拠宣言

#### **hrMIBCompliance MODULE-COMPLIANCE**

STATUS current

DESCRIPTION

The requirements for conformance to the Host Resources MIB.

MANDATORY-GROUPS { hrSystemGroup, hrStorageGroup, hrDeviceGroup }

#### **OBJECT hrSystemDate**

MIN-ACCESS read-only

DESCRIPTION

Write access is not required.

#### **OBJECT hrSystemInitialLoadDevice**

MIN-ACCESS read-only

DESCRIPTION

Write access is not required.

#### **OBJECT hrSystemInitialLoadParameters**

MIN-ACCESS read-only

DESCRIPTION

Write access is not required.

#### **OBJECT hrStorageSize**

MIN-ACCESS read-only

**DESCRIPTION**

Write access is not required.

**OBJECT hrFSLastFullBackupDate**

MIN-ACCESS read-only

DESCRIPTION Write access is not required.

**OBJECT hrFSLastPartialBackupDate**

MIN-ACCESS read-only

**DESCRIPTION**

Write access is not required.

**GROUP hrSWRunGroup****DESCRIPTION**

The Running Software Group. Implementation of this group is mandatory only when the hrSWRunPerfGroup is implemented.

**OBJECT hrSWRunStatus**

MIN-ACCESS read-only

**DESCRIPTION**

Write access is not required.

**GROUP hrSWRunPerfGroup****DESCRIPTION**

The Running Software Performance Group. Implementation of this group is at the discretion of the implementor.

**GROUP hrSWInstalledGroup****DESCRIPTION**

The Installed Software Group. Implementation of this group is at the discretion of the implementor.

::= { hrMIBCompliances 1 }

**hrSystemGroup OBJECT-GROUP**

OBJECTS { hrSystemUptime, hrSystemDate, hrSystemInitialLoadDevice, hrSystemInitialLoadParameters, hrSystemNumUsers, hrSystemProcesses, hrSystemMaxProcesses }

STATUS current

**DESCRIPTION**

The Host Resources System Group.

::= { hrMIBGroups 1 }

**hrStorageGroup OBJECT-GROUP**

OBJECTS { hrMemorySize, hrStorageIndex, hrStorageType, hrStorageDescr, hrStorageAllocationUnits, hrStorageSize, hrStorageUsed, hrStorageAllocationFailures }

STATUS current

**DESCRIPTION**

The Host Resources Storage Group.

::= { hrMIBGroups 2 }

**hrDeviceGroup OBJECT-GROUP**

OBJECTS { hrDeviceIndex, hrDeviceType, hrDeviceDescr, hrDeviceID, hrDeviceStatus, hrDeviceErrors, hrProcessorFrwID, hrProcessorLoad, hrNetworkIfIndex, hrPrinterStatus, hrPrinterDetectedErrorState, hrDiskStorageAccess, hrDiskStorageMedia, hrDiskStorageRemoveble, hrDiskStorageCapacity, hrPartitionIndex, hrPartitionLabel, hrPartitionID, hrPartitionSize, hrPartitionFSIndex, hrFSIndex, hrFSMountPoint, hrFSRemoteMountPoint, hrFSType, hrFSAccess, hrFSBootable, hrFSStorageIndex, hrFSLastFullBackupDate, hrFSLastPartialBackupDate }

STATUS current

DESCRIPTION

The Host Resources Device Group.

::= { hrMIBGroups 3 }

**hrSWRunGroup OBJECT-GROUP**

OBJECTS { hrSWOSIndex, hrSWRunIndex, hrSWRunName, hrSWRunID, hrSWRunPath, hrSWRunParameters, hrSWRunType, hrSWRunStatus }

STATUS current

DESCRIPTION

The Host Resources Running Software Group.

::= { hrMIBGroups 4 }

**hrSWRunPerfGroup OBJECT-GROUP**

OBJECTS { hrSWRunPerfCPU, hrSWRunPerfMem }

STATUS current

DESCRIPTION

The Host Resources Running Software Performance Group.

::= { hrMIBGroups 5 }

**hrSWInstalledGroup OBJECT-GROUP**

OBJECTS { hrSWInstalledLastChange, hrSWInstalledLastUpdateTime, hrSWInstalledIndex, hrSWInstalledName, hrSWInstalledID, hrSWInstalledType, hrSWInstalledDate }

STATUS current

DESCRIPTION

The Host Resources Installed Software Group.

::= { hrMIBGroups 6 }

## Cisco Unified CM Release 6.x の機能サービス

表 8-3 に、Cisco Unified Communications Manager Release 6.x の Cisco Unified Serviceability 機能サービスの一覧を示します。また、適用可能な HOST-RESOURCES-MIB OID、クリア値、およびオブジェクト応答も示します。

表 8-3 Cisco Unified CM Release 6.x の機能サービスと HOST-RESOURCES-MIB

Cisco Unified CM Release 6.x の機能サービス	hrSWRunName OID	クリア値 (肯定ストリング)	オブジェクト応答
Cisco Unified CM Attendant Console Server サービス	1.3.6.1.2.1.25.4.2.1.2	acserver	Cisco CallManager Attendant Console Server サービスの障害
Cisco Extended Functions サービス		cef	Cisco Extended Functions サービスの障害
Cisco Serviceability Reporter サービス		rtmtreporter	Cisco Serviceability Reporter サービスの障害
Compaq Insite Manager サービス		cmascsid	Compaq Insite Manager サービスの障害
Cisco Messaging Interface サービス		cmi	Cisco Messaging Interface サービスの障害
CSA サービス		ciscosecd	Cisco Security Agent サービスの障害
システムでの CISCO-CCM-MIB アクティベーション	1.3.6.1.4.1.9.9.156	ccmAgnt	CCM MIB クエリー機能が無効
IP Voice Media Streaming サービス (アクティブな場合)	1.3.6.1.2.1.25.4.2.1.2	ipvmsd	IP Voice Media Streaming サービスの障害
Cisco Unified CM サービス (アクティブな場合)		ccm	Cisco CallManager サービスの障害
TFTP サービス (アクティブな場合)		ctftp	TFTP サービスの障害
CTIManager サービス (アクティブな場合)		CTIManager	CTIManager サービスの障害
Syslog サービス		syslogd	Syslog サービスの障害
DHCP Monitor サービス (アクティブな場合)		DHCP Monitor	DHCP Monitor サービスの障害
Certificate Trust List サービスの可用性 (アクティブな場合)		CTLProvider	CTLProvider サービスの障害
Certificate Authority Proxy Function サービスの可用性 (アクティブな場合)		capf	Certificate Authority Proxy Function の障害
DirSync サービスの可用性 (アクティブな場合)		CCMDirSync	CCMDirSync サービスの障害
システムでの HOST-RESOURCES MIB アクティベーション	1.3.6.1.2.1.25	host_agent.pl	ホスト MIB クエリー機能が無効

表 8-3 Cisco Unified CM Release 6.x の機能サービスと HOST-RESOURCES-MIB (続き)

Cisco Unified CM Release 6.x の機能サービス	hrSWRunName OID	クリア値 (肯定ストリング)	オブジェクト応答
システムでの MIB2 (RFC1213) アクティベーション	1.3.6.1.2.1	mib2_agent.pl	MIB2 MIB クエリー機能が無効
システムでの SYSAPPL-MIB アクティベーション	1.3.6.1.2.1.54	sapp_agent.pl	SysApp MIB クエリー機能が無効

## Cisco Unified CM Release 6.x のネットワーク サービス

表 8-4 に、Cisco Unified Communications Manager Release 6.x の Cisco Unified Serviceability ネットワーク サービスの一覧を示します。また、適用可能な HOST-RESOURCES-MIB OID、クリア値、およびオブジェクト応答も示します。

表 8-4 Cisco Unified CM Release 6.x のネットワーク サービスと HOST-RESOURCES-MIB

Cisco Unified CM Release 6.x のネットワーク サービス	hrSWRunName OID	クリア値 (肯定ストリング)	オブジェクト応答
Cisco AMC Service サービス	1.3.6.1.2.1.25.4.2.1.2	amc	Cisco AMC Service サービスの障害
Cisco CAR Scheduler サービス		carschlr	Cisco CAR Scheduler サービスの障害
Cisco Trace Collection サービス		tracecollection	Cisco Trace Collection サービスの障害
システムでの HOST-RESOURCES MIB アクティベーション		hostagt	ホスト MIB クエリー機能が無効
システムでの SYSAPPL-MIB アクティベーション	1.3.6.1.2.1.54	sappagt	SysApp MIB クエリー機能が無効
システムでの MIB2 (RFC1213) アクティベーション	1.3.6.1.2.1	mib2agt	MIB2 MIB クエリー機能が無効
システムでの SNMP アクティベーション	1.3.6.1.2.1.25.4.2.1.2	snmp_master_age	システムの SNMP 機能が無効
システムでの SNMP アクティベーション		snmpd	SNMP 機能が無効
システムでの Native Agent Adaptor アクティベーション		naaagt	Native Adaptor Agent 機能が無効
RIS Data Collector サービス		RisDC	RIS Data Collector サービスの障害
CDR Agent サービス		cdragent	CDR Agent サービスの障害

表 8-4 Cisco Unified CM Release 6.x のネットワーク サービスと HOST-RESOURCES-MIB (続き)

Cisco Unified CM Release 6.x のネットワーク サービス	hrSWRunName OID	クリア値 (肯定ストリング)	オブジェクト応答
CDR Replication サービス		cdrrep	CDR Replication サービスの障害
Database Layer Replication サービス		dblrpc	Database Layer Replication サービスの障害
Database Layer Monitor サービス		dbmon	Database Layer Monitor サービスの障害
SSH サービス		sshd	SSH サービスの障害
Syslog サービス		syslogd	Syslog サービスの障害
License Manager サービス		CiscoLicenseMgr	License Manager サービスの障害
System Backup Master サービス		CiscoDRFMaster	System Backup Master サービスの障害
System Backup Local サービス		CiscoDRFLocal	System Backup Local サービスの障害
システムでの CISCO-CDP-MIB アクティベーション	1.3.6.1.4.1.9.9.23	cdpAgt	CDP MIB クエリー機能が無効
CDP サービス		cdpd	CDP サービスの障害
Certificate Expiry Monitor サービスの可用性	1.3.6.1.2.1.25.4.2.1.2	certM	Certificate Expiry Monitor サービスの障害
Syslog サービス		CiscoSyslogSubA	Syslog サービスの障害
データベース サービス		cmoninit	
システムでの HOST-RESOURCES MIB アクティベーション	1.3.6.1.2.1.25	host_agent.pl	ホスト MIB クエリー機能が無効
Tomcat サービス		tomcat	Tomcat サービスの障害
Log Partition Monitoring Tool サービス		LpmTool	Log Partition Monitoring Tool サービスの障害
システムでの SNMP アクティベーション		snmpdm	システムの SNMP 機能が無効

## トラブルシューティング

トラブルシューティングを行うには、次のログと情報を収集する必要があります。

- hostagt ログ ファイル。 **file get activelog /platform/snmp/hostagt/** コマンドを実行して収集します。
- syslog ファイル。 **file get activelog /syslog/** コマンドを実行して収集します。
- Master SNMP Agent ログ ファイル。 **file get activelog /platform/snmp/snmpdm/** コマンドを実行して収集します。
- 実行した一連の操作。

## FAQ

**HOST-RESOURCES-MIB をプロセスのモニタリングに使用できますか。**

HOST-RESOURCES-MIB では、システムで実行されているプロセスに関する情報を hrSwRunTable から取得するために、システムで実行されているすべてのプロセスがモニタされます。インストールされているシスコ製アプリケーションだけをモニタする場合は、SYSAPPL-MIB を使用します。

**HOST-RESOURCES-MIB にマッピングされた RTMT では、メモリ使用率の値はどのように表示されますか。**

表 8-5 に、メモリ使用率の値の一覧を示します。

表 8-5 メモリ使用率の値

メモリ使用率	RTMT カウンタ	HOST-RESOURCES-MIB
スワップメモリの使用率	メモリ\使用されているスワップのキロバイト数	hrStorageUsed.2(この説明は仮想メモリになっています)
物理メモリの使用率	メモリ\使用されているキロバイト数	hrStorageUsed.1(この説明は物理 RAM になっています)
物理メモリとスワップメモリの使用率の合計	メモリ\使用されている仮想メモリのキロバイト数	<p>実際の値ではありません。基本的には、hrStorageUsed.2 と hrStorageUsed.1 を足し合わせる必要があります。</p> <p>負荷の少ないサーバではスワップメモリがまったく使用されない場合があるため、HR 仮想メモリから 0 が返される可能性があります。HR 仮想メモリから正しい値が返されていることを検証するには、この値を RTMT の「メモリ\使用されているスワップのキロバイト数」と比較する必要があります。</p> <p>物理メモリの hrStorageUsed は、使用済みに関するデータ（バッファ+キャッシュ）を示します。</p> <p>MIB で公開されている共有メモリの情報は、HOST-RESOURCES-MIB::hrStorageDescr.10 = STRING: /dev/shm です。HOST-RESOURCES-MIB から報告される仮想メモリは、RTMT ではスワップメモリと見なされているものです。</p> <p>HOST-RESOURCES-MIB では、次の式が使用されます。</p> <ul style="list-style-type: none"> <li>• %物理メモリの使用率 = (物理 RAM の hrStorageUsed + /dev/shm hrStorageUsed) / (物理 RAM の hrStorageSize)</li> <li>• %使用されている仮想メモリ = (物理 RAM の hrStorageUsed + /dev/shm hrStorageUsed + 仮想メモリの hrStorageUsed) / (物理 RAM の hrStorageSize + 仮想メモリの hrStorageSize)</li> </ul>

**RTMT で表示されるディスク容量の値と HOST-RESOURCES-MIB のディスク容量の値が異なるのはなぜですか。**

一般に、df サイズでは、表示される使用済みディスク容量や使用可能ディスク容量の値は一致しません。これは、ファイルシステムの予約済みディスクブロックにおける最小空き容量の割合が原因です。Releases 6.x および 7.0 システムでの Cisco Unified Communication Manager の最小空き容量の値は 1% です。このため、表示される使用済みディスク容量の値には、RTMT と HOST-RESOURCES-MIB との間で 1% の差異が生じるようになります。



RTMT では、df の報告値を使用して使用済みディスク容量の値が表示されます。この値は、 $[(\text{合計容量} - \text{使用可能容量}) / \text{合計容量}] \times 100$  で計算されます。合計容量には最小空き容量も含まれます。HOST-RESOURCES-MIB でのこの値は、 $[\text{hrStorageUsed} / \text{hrStorageSize}] \times 100$  で計算されます。hrStorageSize には最小空き容量は含まれません。

### hrStorageUsed の値は、ホスト エージェントではどのように表示されますか。

物理 RAM の hrStorageUsed は、使用済みに関するデータ (バッファ + キャッシュ) を表示するために修正されました。ホスト エージェントのバージョンが正しいかどうかを確認するには、show packages active snmp コマンドを使用して、システムにインストールされている snmp-rpm のバージョンを収集してください。

メモリ容量およびメモリ使用率の値を HOST-RESOURCES-MIB での値と比較する方法について教えてください。

HOST-RESOURCES-MIB では、使用済みのサイズとストレージが hrStorageUnits 単位で表されます。対象のストレージタイプにおける hrStorageUnits が 4096 バイトの場合、MIB 値で照会された hrStorageUsed または hrStorageSize の値に 4096 を掛ける必要があります。たとえば、show status コマンドでは、物理 RAM の合計メモリが 4090068K と表示されます。

physicalRAM ストレージタイプの hrStorageUnits が 4096 バイトの場合、物理 RAM の hrStorageSize は 1022517 と表示されます。これは、4090078K に相当します  $((1022517 \times 4096) / 1024 = 4090068K)$ 。

### Windows の場合、HOST-RESOURCES-MIB の hrSWRunName に対する SNMP クエリーでは、不正なエントリが断続的に返されます。

Microsoft 社の SNMP 拡張エージェント (hostmib.dll) では、HOST-RESOURCE-MIB がサポートされています。このため、Microsoft 社のサポートがこの問題の解決に役立つ場合があります。問題が解消されない場合、次の点について確認することを推奨します。

- tlist snmp.exe ファイルを使用して、hostmib.dll が出力に表示されることを確認する。
- SNMP サービスの開始時に、SNMP からのエラー メッセージまたは警告メッセージがイベントビューアに表示されていないことを確認する。
- 使用されているコミュニティストリングが、snmp サービス プロパティで読み取り権限付きで設定されていることを確認する。
- MSSQL-MIB (MssqlSrvInfoTable) を使用して、sql プロセスのステータスを確認する。

### プロセスのモニタリング

HOST-RESOURCES-MIB は、システムで実行されているすべてのプロセスに関する情報を hrSWRunTable から取得します。システムで実行されているすべてのプロセスをモニタする場合は、この MIB を使用します。インストールされているシスコ製アプリケーションだけをモニタする場合は、SYSAPPL-MIB.Disk Space および RTMT を使用します。

HOST-RESOURCES-MIB で表示される使用済みディスク容量と使用可能ディスク容量の値は、RTMT で表示されるディスク容量の値と一致しない場合があります。これは、ファイルシステムの予約済みディスク ブロックにおける最小空き容量の割合が原因です。6.x および 7.0 システムでの Cisco Unified Communications Manager の最小空き容量の値は 1% であるため、表示される使用済みディスク容量の値には、RTMT と HOST-RESOURCES-MIB との間で 1% の差異が生じるようになります。

- RTMT では、df の報告値を使用して使用済みディスク容量の値が表示されます。この値は、 $[(\text{合計容量} - \text{使用可能容量}) / \text{合計容量}] \times 100$  で計算されます。合計容量には最小空き容量も含まれます。
- HOST-RESOURCES-MIB での使用済みディスク容量の値は  $[\text{hrStorageUsed} / \text{hrStorageSize}] \times 100$  で計算されます。hrStorageSize には最小空き容量は含まれません。

# IF-MIB



(注)

この IF-MIB は形式が変更されています。この項のすべての MIB は、<http://tools.cisco.com/Support/SNMP/do/BrowseMIB.do?local=en&step=2> からダウンロードしてコンパイルしてください。

IF-MIB をコンパイルするには、次に示されている順番で MIB をコンパイルする必要があります。

1. SNMPv2-SMI
2. SNMPv2-TC
3. SNMPv2-CONF
4. SNMPv2-MIB
5. IANAifType-MIB
6. RFC1155-SMI
7. RFC-1212
8. SNMPv2-SMI-v1
9. RFC-1215
10. SNMPv2-TC-v1
11. IF-MIB

さらに、次のファイルをダウンロードします。

- OID ファイル : IF-MIB.oid

この項の内容は次のとおりです。

- 「改訂」 (P.8-107)
- 「定義」 (P.8-107)
- 「オブジェクト」 (P.8-107)
- 「テキストの表記法」 (P.8-107)
- 「インターフェイス インデックス」 (P.8-108)
- 「インターフェイス テーブル」 (P.8-109)
- 「インターフェイス テーブルへの拡張」 (P.8-116)
- 「高キャパシティ カウンタ オブジェクト」 (P.8-118)
- 「インターフェイス スタック グループ」 (P.8-122)
- 「汎用受信アドレス テーブル」 (P.8-123)
- 「インターフェイス関連のトラップの定義」 (P.8-125)
- 「適合情報」 (P.8-125)
- 「準拠宣言」 (P.8-125)
- 「適合単位」 (P.8-127)
- 「非推奨の定義 : オブジェクト」 (P.8-129)
- 「非推奨の定義 : グループ」 (P.8-133)

- 「非推奨の定義：準拠性」(P.8-134)

## 改訂

表 8-2 に、最新の改訂から順に MIB の改訂を示します。

表 8-6 改訂の履歴

日付	処置	説明
2000/06/14	更新しました	ネットワーク インターフェイス サブレイヤ用の汎用オブジェクトについて記述する MIB モジュール。この MIB は、MIB-II の ifTable のアップデートバージョンであり、RFC 1229 で定義されている拡張機能を備えています。  インターフェイス MIB WG によって取り決められた詳細事項であり、RFC 2863 として公開されています。
1996/02/28	改訂しました	インターフェイス MIB WG によって行われた改訂であり、RFC 2233 で公開されています。
1993/08/11	初版	RFC 1573 の一部として公開されています。  ::= {mib-2 31}

## 定義

次の定義が IF-MIB 用にインポートされています。

- MODULE-IDENTITY、OBJECT-TYPE、Counter32、Gauge32、Counter64、Integer32、TimeTicks、mib-2、NOTIFICATION-TYPE
- SNMPv2-SMI から：TEXTUAL-CONVENTION、DisplayString、PhysAddress、TruthValue、RowStatus、TimeStamp、AutonomousType、TestAndIncr
- SNMPv2-TC から：MODULE-COMPLIANCE、OBJECT-GROUP、NOTIFICATION-GROUP
- SNMPv2-CONF から：snmpTraps
- SNMPv2-MIB から：IANAifType
- IANAifType-MIB から

## オブジェクト

**ifMIBObjects** OBJECT IDENTIFIER ::= {ifMIB 1}

**interfaces** OBJECT IDENTIFIER ::= {mib-2 2}

## テキストの表記法



(注) OwnerString のセマンティックは、RFC 1271 で使用されているものと同じです。

**OwnerString** ::= TEXTUAL-CONVENTION

DISPLAY-HINT 255a

STATUS deprecated

DESCRIPTION

This data type is used to model an administratively assigned name of the owner of a resource. This information is taken from the NVT ASCII character set. It is suggested that this name contain one or more of the following: ASCII form of the manager station's transport address, management station name (e.g., domain name), network management personnel's name, location, or phone number. In some cases the agent itself will be the owner of an entry. In these cases, this string shall be set to a string starting with agent.

A value which indicates the set of services that this entity may potentially offers. The value is a sum. This sum initially takes the value zero, Then, for each layer, L, in the range 1 through 7, that this node performs transactions for,  $2^{L-1}$  is added to the sum. For example, a node which performs only routing functions would have a value of 4 ( $2^{(3-1)}$ ). In contrast, a node which is a host offering application services would have a value of 72 ( $2^{(4-1)} + 2^{(7-1)}$ ). Note that in the context of the Internet suite of protocols, values should be calculated accordingly:

Layer functionality:

- 1—physical (e.g., repeaters)
- 2—datalink/subnetwork (e.g., bridges)
- 3—internet (e.g., supports the IP)
- 4—end-to-end (e.g., supports the TCP)
- 7—applications (e.g., supports the SMTP)

For systems including OSI protocols, layers 5 and 6 may also be counted.

SYNTAX Octet String (SIZE(0..255))

## インターフェイス インデックス

インターフェイス インデックスには、ifIndex のセマンティックが格納されます。他の MIB モジュールで定義されたオブジェクトでこれらのセマンティックが必要な場合は、インターフェイス インデックスを使用する必要があります。

**InterfaceIndex ::= TEXTUAL-CONVENTION**

DISPLAY-HINT d

STATUS current

DESCRIPTION

A unique value, greater than zero, for each interface or interface sub-layer in the managed system. It is recommended that values are assigned contiguously starting from 1. The value for each interface sub-layer must remain constant at least from one re-initialization of the entity's network management system to the next re-initialization.

SYNTAX Integer32 (1..2147483647)

**InterfaceIndexOrZero ::= TEXTUAL-CONVENTION**

DISPLAY-HINT d

STATUS current

DESCRIPTION

This textual convention is an extension of the InterfaceIndex convention. The latter defines a greater than zero value used to identify an interface or interface sub-layer in the managed system. This extension permits the additional value of zero. 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 interface was unknown, or when none or all interfaces need to be referenced.

SYNTAX Integer32 (0..2147483647)

**ifNumber OBJECT-TYPE**

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of network interfaces (regardless of their current state) present on this system.

::= { interfaces 1 }

**ifTableLastChange 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 ifTable. If the number of entries has been unchanged since the last re-initialization of the local network management subsystem, then this object contains a zero value.

::= { ifMIBObjects 5 }

## インターフェイス テーブル

インターフェイス テーブルには、エンティティのインターフェイスに関する情報が格納されます。ネットワーク インターフェイスのインターネットワーク レイヤの下位にある各サブレイヤは、インターフェイスと見なされます。

**ifTable OBJECT-TYPE**

SYNTAX Sequence of IfEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

A list of interface entries. The number of entries is given by the value of ifNumber.

::= { interfaces 2 }

**ifEntry OBJECT-TYPE**

SYNTAX IfEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

An entry containing management information applicable to a particular interface.

INDEX {ifIndex}

::= {ifTable 1}

IfEntry ::=

SEQUENCE {ifIndex InterfaceIndex, ifDescr DisplayString, ifType IANAifType, ifMtu Integer32, filespec Gauge32, ifPhysAddress PhysAddress, ifAdminStatus INTEGER, ifOperStatusINTEGER, ifLastChangeTimeTicks, ifInOctets Counter32, ifInUcastPkts Counter32, ifInNUcastPkts Counter32, -- deprecated ifInDiscardsCounter32, ifInErrors Counter32, ifInUnknownProtos Counter32, ifOutOctets Counter32, ifOutUcastPkts Counter32, ifOutNUcastPkts Counter32, -- deprecated ifOutDiscards Counter32, ifOutErrors Counter32, ifOutQLen Gauge32,-- deprecated ifSpecific OBJECT IDENTIFIER -- deprecated}

**ifIndex OBJECT-TYPE**

SYNTAX InterfaceIndex

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

A unique value, greater than zero, for each interface. It is recommended that values are assigned contiguously starting from 1. The value for each interface sub-layer must remain constant at least from one re-initialization of the entity's network management system to the next re-initialization.

::= {ifEntry 1}

**ifDescr OBJECT-TYPE**

SYNTAX DisplayString (SIZE (0..255))

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

A textual string containing information about the interface. This string should include the name of the manufacturer, the product name and the version of the interface hardware/software.

::= {ifEntry 2}

**ifType OBJECT-TYPE**

SYNTAX IANAifType

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

The type of interface. Additional values for ifType are assigned by the Internet Assigned Numbers Authority (IANA), through updating the syntax of the IANAifType textual convention.

::= {ifEntry 3}

**ifMtu OBJECT-TYPE**

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The size of the largest packet which can be sent/received on the interface, specified in octets. For interfaces that are used for transmitting network datagrams, this is the size of the largest network datagram that can be sent on the interface.

::= {ifEntry 4}

#### **ifSpeed OBJECT-TYPE**

SYNTAX Gauge32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

An estimate of the interface current bandwidth in bits per second. For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth. If the bandwidth of the interface is greater than the maximum value reportable by this object then this object should report its maximum value (4,294,967,295) and ifHighSpeed must be used to report the interface speed. For a sub-layer which has no concept of bandwidth, this object should be zero.

::= {ifEntry 5}

#### **ifPhysAddress OBJECT-TYPE**

SYNTAX PhysAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The interface's address at its protocol sub-layer. For example, for an 802.x interface, this object normally contains a MAC address. The interface's media-specific MIB must define the bit and byte ordering and the format of the value of this object. For interfaces which do not have such an address (e.g., a serial line), this object should contain an octet string of zero length.

::= {ifEntry 6}

#### **ifAdminStatus OBJECT-TYPE**

SYNTAX Integer {up(1), -- ready to pass packets down(2), testing(3) -- in some test mode}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

The desired state of the interface. The testing(3) state indicates that no operational packets can be passed. When a managed system initializes, all interfaces start with ifAdminStatus in the down(2) state. As a result of either explicit management action or per configuration information retained by the managed system, ifAdminStatus is then changed to either the up(1) or testing(3) states (or remains in the down(2) state).

::= {if Entry 7}

#### **ifOperStatus OBJECT-TYPE**

SYNTAX INTEGER {up(1),-- ready to pass packets down(2), testing(3), -- in some test mode unknown(4), -- status can not be determined -- for some reason.dormant(5), notPresent(6),-- some component is missing lowerLayerDown(7) -- down due to state of -- lower-layer interface(s)}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The current operational state of the interface.The testing(3) state indicates that no operational packets can be passed.If ifAdminStatus is down(2) then ifOperStatus should be down(2).If ifAdminStatus is changed to up(1) then ifOperStatus should change to up(1) if the interface is ready to transmit and receive network traffic; it should change to dormant(5) if the interface is waiting for external actions (such as a serial line waiting for an incoming connection); it should remain in the down(2) state if and only if there is a fault that prevents it from going to the up(1) state; it should remain in the notPresent(6) state if the interface has missing (typically, hardware) components.

::= {ifEntry 8}

#### **ifLastChange OBJECT-TYPE**

SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The value of sysUpTime at the time the interface entered its current operational state.If the current state was entered prior to the last re-initialization of the local network management subsystem, then this object contains a zero value.

::= {ifEntry 9}

#### **ifInOctets OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The total number of octets received on the interface, including framing characters.Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

::= { ifEntry 10 }

#### **ifInUcastPkts OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were not addressed to a multicast or broadcast address at this sub-layer.Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

::= { ifEntry 11 }



**ifInNUcastPkts OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a multicast or broadcast address at this sub-layer. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

This object is deprecated in favour of ifInMulticastPkts and ifInBroadcastPkts.

::= { ifEntry 12 }

**ifInDiscards OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

::= { ifEntry 13 }

**ifInErrors OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

::= { ifEntry 14 }

**ifInUnknownProtos OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

For packet-oriented interfaces, the number of packets received via the interface which were discarded because of an unknown or unsupported protocol. For character-oriented or fixed-length interfaces that support protocol multiplexing the number of transmission units received via the interface which were discarded because of an unknown or unsupported protocol. For any interface that does not support protocol multiplexing, this counter will always be 0.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

::= { ifEntry 15 }

#### **ifOutOctets OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The total number of octets transmitted out of the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

::= { ifEntry 16 }

#### **ifOutUcastPkts OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The total number of packets that higher-level protocols requested be transmitted, and which were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

::= { ifEntry 17 }

#### **ifOutNUcastPkts OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

This object is deprecated in favour of ifOutMulticastPkts and ifOutBroadcastPkts.

::= { ifEntry 18 }

#### **ifOutDiscards OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

::= { ifEntry 19 }

**ifOutErrors OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

::= { ifEntry 20 }

**ifOutQLen OBJECT-TYPE**

SYNTAX Gauge32

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

The length of the output packet queue (in packets).

::= { ifEntry 21 }

**ifSpecific OBJECT-TYPE**

SYNTAX OBJECT IDENTIFIER

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

A reference to MIB definitions specific to the particular media being used to realize the interface. It is recommended that this value point to an instance of a MIB object in the media-specific MIB, i.e., that this object have the semantics associated with the InstancePointer textual convention defined in RFC 2579. In fact, it is recommended that the media-specific MIB specify what value ifSpecific should/can take for values of ifType. If no MIB definitions specific to the particular media are available, the value should be set to the OBJECT IDENTIFIER { 0 0 }.

::= { ifEntry 22 }

## インターフェイス テーブルへの拡張

このテーブルは、ifExtnsTable テーブルに置き換わるものです。

### ifXTableOBJECT-TYPE

SYNTAX Sequence of IfXEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

A list of interface entries. The number of entries is given by the value of ifNumber. This table contains additional objects for the interface table.

::= { ifMIBObjects 1 }

### ifXEntryOBJECT-TYPE

SYNTAX IfXEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry containing additional management information applicable to a particular interface.

AUGMENTS { ifEntry }

::= { ifXTable 1 }

IfXEntry ::=

SEQUENCE { ifName DisplayString, ifInMulticastPkts Counter32, ifInBroadcastPkts Counter32, ifOutMulticastPkts Counter32, ifOutBroadcastPkts Counter32, ifHCInOctetsCounter64, ifHCInUcastPkts Counter64, ifHCInMulticastPkts Counter64, ifHCInBroadcastPkts Counter64, ifHCOutOctets Counter64, ifHCOutUcastPktsCounter64, ifHCOutMulticastPktsCounter64, ifHCOutBroadcastPktsCounter64, ifLinkUpDownTrapEnable INTEGER, ifHighSpeed Gauge32, ifPromiscuousMode TruthValue, ifConnectorPresent TruthValue, ifAlias DisplayString, ifCounterDiscontinuityTime TimeStamp }

### ifName OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The textual name of the interface. The value of this object should be the name of the interface as assigned by the local device and should be suitable for use in commands entered at the device's 'console'. This might be a text name, such as 'le0' or a simple port number, such as '1', depending on the interface naming syntax of the device. If several entries in the ifTable together represent a single interface as named by the device, then each will have the same value of ifName. Note that for an agent which responds to SNMP queries concerning an interface on some other (proxied) device, then the value of ifName for such an interface is the proxied device's local name for it.

If there is no local name, or this object is otherwise not applicable, then this object contains a zero-length string.

::= { ifXEntry 1 }

**ifInMulticastPkts OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a multicast address at this sub-layer. For a MAC layer protocol, this includes both Group and Functional addresses. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

::= { ifXEntry 2 }

**ifInBroadcastPkts OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a broadcast address at this sub-layer. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

::= { ifXEntry 3 }

**ifOutMulticastPkts OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a multicast address at this sub-layer, including those that were discarded or not sent. For a MAC layer protocol, this includes both Group and Functional addresses.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

::= { ifXEntry 4 }

**ifOutBroadcastPkts OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a broadcast address at this sub-layer, including those that were discarded or not sent.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

::= { ifXEntry 5 }

## 高キャパシティ カウンタ オブジェクト

これらのオブジェクトはすべて、基本的な ifTable カウンタの 64 ビットバージョンです。これらのすべてのオブジェクトの基本的なセマンティックは、32 ビットバージョンと同じですが、その構文が 64 ビットに拡張されています。

### ifHCInOctets OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The total number of octets received on the interface, including framing characters. This object is a 64-bit version of ifInOctets. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

::= { ifXEntry 6 }

### ifHCInUcastPkts OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were not addressed to a multicast or broadcast address at this sub-layer. This object is a 64-bit version of ifInUcastPkts.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

::= { ifXEntry 7 }

### ifHCInMulticastPkts OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a multicast address at this sub-layer. For a MAC layer protocol, this includes both Group and Functional addresses. This object is a 64-bit version of ifInMulticastPkts.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

::= { ifXEntry 8 }

### ifHCInBroadcastPkts OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a broadcast address at this sub-layer. This object is a 64-bit version of ifInBroadcastPkts.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

::= { ifXEntry 9 }

#### **ifHCOctets OBJECT-TYPE**

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The total number of octets transmitted out of the interface, including framing characters. This object is a 64-bit version of ifOutOctets.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

::= { ifXEntry 10 }

#### **ifHCOucastPkts OBJECT-TYPE**

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The total number of packets that higher-level protocols requested be transmitted, and which were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. This object is a 64-bit version of ifOutUcastPkts.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

::= { ifXEntry 11 }

#### **ifHCOmulticastPkts OBJECT-TYPE**

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a multicast address at this sub-layer, including those that were discarded or not sent. For a MAC layer protocol, this includes both Group and Functional addresses. This object is a 64-bit version of ifOutMulticastPkts.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

::= { ifXEntry 12 }

#### **ifHCObroadcastPkts OBJECT-TYPE**

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a broadcast address at this sub-layer, including those that were discarded or not sent. This object is a 64-bit version of ifOutBroadcastPkts.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

::= { ifXEntry 13 }

#### **ifLinkUpDownTrapEnable OBJECT-TYPE**

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

MAX-ACCESS read-write

STATUS current

DESCRIPTION

Indicates whether linkUp/linkDown traps should be generated for this interface. By default, this object should have the value enabled(1) for interfaces which do not operate on 'top' of any other interface (as defined in the ifStackTable), and disabled(2) otherwise.

::= { ifXEntry 14 }

#### **ifHighSpeed OBJECT-TYPE**

SYNTAX Gauge32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

An estimate of the interface's current bandwidth in units of 1,000,000 bits per second. If this object reports a value of 'n' then the speed of the interface is somewhere in the range of 'n-500,000' to 'n+499,999'. For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth. For a sub-layer which has no concept of bandwidth, this object should be zero.

::= { ifXEntry 15 }

#### **ifPromiscuousMode OBJECT-TYPE**

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

This object has a value of false(2) if this interface only accepts packets/frames that are addressed to this station. This object has a value of true(1) when the station accepts all packets/frames transmitted on the media. The value true(1) is only legal on certain types of media. If legal, setting this object to a value of true(1) may require the interface to be reset before becoming effective.

The value of ifPromiscuousMode does not affect the reception of broadcast and multicast packets/frames by the interface.

::= { ifXEntry 16 }

#### **ifConnectorPresent OBJECT-TYPE**



SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object has the value 'true(1)' if the interface sublayer has a physical connector and the value 'false(2)' otherwise.

::= { ifXEntry 17 }

#### **ifAlias OBJECT-TYPE**

SYNTAX DisplayString (SIZE(0..64))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

This object is an alias name for the interface as specified by a network manager, and provides a non-volatile handle for the interface.

On the first instantiation of an interface, the value of ifAlias associated with that interface is the zero-length string. As and when a value is written into an instance of ifAlias through a network management set operation, then the agent must retain the supplied value in the ifAlias instance associated with the same interface for as long as that interface remains instantiated, including across all re-initializations/reboots of the network management system, including those which result in a change of the interface's ifIndex value.

An example of the value which a network manager might store in this object for a WAN interface is the (Telco's) circuit number/identifier of the interface.

Some agents may support write-access only for interfaces having particular values of ifType. An agent which supports write access to this object is required to keep the value in non-volatile storage, but it may limit the length of new values depending on how much storage is already occupied by the current values for other interfaces.

::= { ifXEntry 18 }

#### **ifCounterDiscontinuityTime OBJECT-TYPE**

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The value of sysUpTime on the most recent occasion at which any one or more of this interface's counters suffered a discontinuity. The relevant counters are the specific instances associated with this interface of any Counter32 or Counter64 object contained in the ifTable or ifXTable. If no such discontinuities have occurred since the last re-initialization of the local management subsystem, then this object contains a zero value.

::= { ifXEntry 19 }

## インターフェイス スタック グループ

このグループの実装は省略可能ですが、すべてのシステムに実装することを強く推奨します。

### ifStackTable OBJECT-TYPE

SYNTAX Sequence of IfStackEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table containing information on the relationships between the multiple sub-layers of network interfaces. In particular, it contains information on which sub-layers run 'on top of' which other sub-layers, where each sub-layer corresponds to a conceptual row in the ifTable. For example, when the sub-layer with ifIndex value x runs over the sub-layer with ifIndex value y, then this table contains ifStackStatus.x.y=active.

For each ifIndex value, I, which identifies an active interface, there are always at least two instantiated rows in this table associated with I. For one of these rows, I is the value of ifStackHigherLayer; for the other, I is the value of ifStackLowerLayer. (If I is not involved in multiplexing, then these are the only two rows associated with I.)

For example, two rows exist even for an interface which has no others stacked on top or below it:

- ifStackStatus.0.x=active
- ifStackStatus.x.0=active

::= { ifMIBObjects 2 }

### ifStackEntry OBJECT-TYPE

SYNTAX IfStackEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

Information on a particular relationship between two sub-layers, specifying that one sub-layer runs on 'top' of the other sub-layer. Each sub-layer corresponds to a conceptual row in the ifTable.

INDEX { ifStackHigherLayer, ifStackLowerLayer }

::= { ifStackTable 1 }

IfStackEntry ::= SEQUENCE { ifStackHigherLayer InterfaceIndexOrZero, ifStackLowerLayer InterfaceIndexOrZero, ifStackStatus RowStatus }

### ifStackHigherLayer OBJECT-TYPE

SYNTAX InterfaceIndexOrZero

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The value of ifIndex corresponding to the higher sub-layer of the relationship, i.e., the sub-layer which runs on 'top' of the sub-layer identified by the corresponding instance of ifStackLowerLayer. If there is no higher sub-layer (below the internetwork layer), then this object has the value 0.

::= { ifStackEntry 1 }

**ifStackLowerLayer OBJECT-TYPE**

SYNTAX InterfaceIndexOrZero

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The value of ifIndex corresponding to the lower sub-layer of the relationship, i.e., the sub-layer which runs 'below' the sub-layer identified by the corresponding instance of ifStackHigherLayer.If there is no lower sub-layer, then this object has the value 0.

::= { ifStackEntry 2 }

**ifStackStatus OBJECT-TYPE**

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

The status of the relationship between two sub-layers.Changing the value of this object from 'active' to 'notInService' or 'destroy' will likely have consequences up and down the interface stack.Thus, write access to this object is likely to be inappropriate for some types of interfaces, and many implementations will choose not to support write-access for any type of interface.

::= { ifStackEntry 3 }

**ifStackLastChange OBJECT-TYPE**

SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The value of sysUpTime at the time of the last change of the (whole) interface stack.A change of the interface stack is defined to be any creation, deletion, or change in value of any instance of ifStackStatus.If the interface stack has been unchanged since the last re-initialization of the local network management subsystem, then this object contains a zero value.

::= { ifMIBObjects 6 }

## 汎用受信アドレス テーブル

このオブジェクト グループは、複数のアドレスにアドレス指定されているパケットまたはフレームを受信できる、すべてのタイプのインターフェイスに必須です。このテーブルは ifExtnsRcvAddr テーブルに代わるものです。主な違いは、このテーブルでは RowStatus のテキストの表記法を使用していますが、ifExtnsRcvAddr ではこの表記法を使用していない点です。

**ifRcvAddressTable OBJECT-TYPE**

SYNTAX Sequence of IfRcvAddressEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

This table contains an entry for each address (broadcast, multicast, or uni-cast) for which the system will receive packets/frames on a particular interface, except as follows:

- For an interface operating in promiscuous mode, entries are only required for those addresses for which the system would receive frames were it not operating in promiscuous mode.
- For 802.5 functional addresses, only one entry is required, for the address which has the functional address bit ANDed with the bit mask of all functional addresses for which the interface will accept frames.

A system is normally able to use any unicast address which corresponds to an entry in this table as a source address.

::= { ifMIBObjects 4 }

#### **ifRcvAddressEntry OBJECT-TYPE**

SYNTAX IfRcvAddressEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

A list of objects identifying an address for which the system will accept packets/frames on the particular interface identified by the index value ifIndex.

INDEX { ifIndex, ifRcvAddressAddress }

::= { ifRcvAddressTable 1 }

IfRcvAddressEntry ::= SEQUENCE { ifRcvAddressAddress PhysAddress,  
ifRcvAddressStatusRowStatus, ifRcvAddressType INTEGER }

#### **ifRcvAddressAddress OBJECT-TYPE**

SYNTAX PhysAddress

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An address for which the system will accept packets/frames on this entry's interface.

::= { ifRcvAddressEntry 1 }

#### **ifRcvAddressStatus OBJECT-TYPE**

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

This object is used to create and delete rows in the ifRcvAddressTable.

::= { ifRcvAddressEntry 2 }

#### **ifRcvAddressType OBJECT-TYPE**

SYNTAX INTEGER { other(1), volatile(2), nonVolatile(3) }

MAX-ACCESS read-create

STATUS current

DESCRIPTION

This object has the value nonVolatile(3) for those entries in the table which are valid and will not be deleted by the next restart of the managed system. Entries having the value volatile(2) are valid and exist, but have not been saved, so that will not exist after the next restart of the managed system. Entries having the value other(1) are valid and exist but are not classified as to whether they will continue to exist after the next restart.

```
DEFVAL { volatile }
::= { ifRcvAddressEntry 3 }
```

## インターフェイス関連のトラップの定義

### linkDown NOTIFICATION-TYPE

```
OBJECTS { ifIndex, ifAdminStatus, ifOperStatus }
STATUS current
DESCRIPTION
```

A linkDown trap signifies that the SNMP entity, acting in an agent role, has detected that the ifOperStatus object for one of its communication links is about to enter the down state from some other state (but not from the notPresent state). This other state is indicated by the included value of ifOperStatus.

```
::= { snmpTraps 3 }
```

### linkUp NOTIFICATION-TYPE

```
OBJECTS { ifIndex, ifAdminStatus, ifOperStatus }
STATUS current
DESCRIPTION
```

A linkUp trap signifies that the SNMP entity, acting in an agent role, has detected that the ifOperStatus object for one of its communication links left the down state and transitioned into some other state (but not into the notPresent state). This other state is indicated by the included value of ifOperStatus.

```
::= { snmpTraps 4 }
```

## 適合情報

```
ifConformance OBJECT IDENTIFIER ::= { ifMIB 2 }
ifGroups OBJECT IDENTIFIER ::= { ifConformance 1 }
ifCompliances OBJECT IDENTIFIER ::= { ifConformance 2 }
```

## 準拠宣言

### ifCompliance3 MODULE-COMPLIANCE

```
STATUS current
DESCRIPTION
```

The compliance statement for SNMP entities which have network interfaces.

MODULE -- this module

MANDATORY-GROUPS { ifGeneralInformationGroup, linkUpDownNotificationsGroup }

The groups:

- ifFixedLengthGroup
- ifHCFixedLengthGroup
- ifPacketGroup
- ifHCPacketGroup
- ifVHCPacketGroup

Mutually exclusive; at most one of these groups is implemented for a particular interface. When any of these groups is implemented for a particular interface, then ifCounterDiscontinuityGroup must also be implemented for that interface.

GROUP ifFixedLengthGroup

DESCRIPTION

This group is mandatory for those network interfaces which are character-oriented or transmit data in fixed-length transmission units, and for which the value of the corresponding instance of ifSpeed is less than or equal to 20,000,000 bits/second.

GROUP ifHCFixedLengthGroup

DESCRIPTION

This group is mandatory for those network interfaces which are character-oriented or transmit data in fixed-length transmission units, and for which the value of the corresponding instance of ifSpeed is greater than 20,000,000 bits/second.

GROUP ifPacketGroup

DESCRIPTION

This group is mandatory for those network interfaces which are packet-oriented, and for which the value of the corresponding instance of ifSpeed is less than or equal to 20,000,000 bits/second.

GROUP ifHCPacketGroup

DESCRIPTION

This group is mandatory only for those network interfaces which are packet-oriented and for which the value of the corresponding instance of ifSpeed is greater than 20,000,000 bits/second but less than or equal to 650,000,000 bits/second.

GROUP ifVHCPacketGroup

DESCRIPTION

This group is mandatory only for those network interfaces which are packet-oriented and for which the value of the corresponding instance of ifSpeed is greater than 650,000,000 bits/second.

GROUP ifCounterDiscontinuityGroup

DESCRIPTION

This group is mandatory for those network interfaces that are required to maintain counters (i.e., those for which one of the ifFixedLengthGroup, ifHCFixedLengthGroup, ifPacketGroup, ifHCPacketGroup, or ifVHCPacketGroup is mandatory).

GROUP ifRcvAddressGroup

DESCRIPTION

The applicability of this group MUST be defined by the media-specific MIBs. Media-specific MIBs must define the exact meaning, use, and semantics of the addresses in this group.

OBJECT ifLinkUpDownTrapEnable

MIN-ACCESS read-only

DESCRIPTION

Write access is not required.

OBJECT ifPromiscuousMode

MIN-ACCESS read-only

DESCRIPTION

Write access is not required.

OBJECT ifAdminStatus

SYNTAX INTEGER { up(1), down(2) }

MIN-ACCESS read-only

DESCRIPTION

Write access is not required, nor is support for the value testing(3).

OBJECT ifAlias

MIN-ACCESS read-only

DESCRIPTION

Write access is not required.

::= { ifCompliances 3 }

## 適合単位

### ifGeneralInformationGroupOBJECT-GROUP

OBJECTS { ifIndex, ifDescr, ifType, ifSpeed, ifPhysAddress, ifAdminStatus, ifOperStatus, ifLastChange, ifLinkUpDownTrapEnable, ifConnectorPresent, ifHighSpeed, ifName, ifNumber, ifAlias, ifTableLastChange }

STATUS current

DESCRIPTION

A collection of objects providing information applicable to all network interfaces.

::= { ifGroups 10 }



(注)

次の 5 つのグループは相互に排他的です。これらのグループの 1 つだけがいずれかのインターフェイスに実装されます。

- ifFixedLengthGroupOBJECT-GROUP

OBJECTS { ifInOctets, ifOutOctets, ifInUnknownProtos, ifInErrors, ifOutErrors }

STATUS current

DESCRIPTION

A collection of objects providing information specific to non-high speed (non-high speed interfaces transmit and receive at speeds less than or equal to 20,000,000 bits/second) character-oriented or fixed-length-transmission network interfaces.

::= { ifGroups 2 }

#### **ifHCFixedLengthGroupOBJECT-GROUP**

OBJECTS { ifHCInOctets, ifHCOctets, ifInOctets, ifOutOctets, ifInUnknownProtos, ifInErrors, ifOutErrors }

STATUS current

DESCRIPTION

A collection of objects providing information specific to high speed (greater than 20,000,000 bits/second) character-oriented or fixed-length-transmission network interfaces.

::= { ifGroups 3 }

#### **ifPacketGroupOBJECT-GROUP**

OBJECTS { ifInOctets, ifOutOctets, ifInUnknownProtos, ifInErrors, ifOutErrors, ifMtu, ifInUcastPkts, ifInMulticastPkts, ifInBroadcastPkts, ifInDiscards, ifOutUcastPkts, ifOutMulticastPkts, ifOutBroadcastPkts, ifOutDiscards, ifPromiscuousMode }

STATUS current

DESCRIPTION

A collection of objects providing information specific to non-high speed (non-high speed interfaces transmit and receive at speeds less than or equal to 20,000,000 bits/second) packet-oriented network interfaces.

::= { ifGroups 4 }

#### **ifHCPacketGroupOBJECT-GROUP**

OBJECTS { ifHCInOctets, ifHCOctets, ifInOctets, ifOutOctets, ifInUnknownProtos, ifInErrors, ifOutErrors, ifMtu, ifInUcastPkts, ifInMulticastPkts, ifInBroadcastPkts, ifInDiscards, ifOutUcastPkts, ifOutMulticastPkts, ifOutBroadcastPkts, ifOutDiscards, ifPromiscuousMode }

STATUS current

DESCRIPTION

A collection of objects providing information specific to high speed (greater than 20,000,000 bits/second but less than or equal to 650,000,000 bits/second) packet-oriented network interfaces.

::= { ifGroups 5 }

#### **ifVHCPacketGroupOBJECT-GROUP**

OBJECTS { ifHCInUcastPkts, ifHCInMulticastPkts, ifHCInBroadcastPkts, ifHCOctets, ifHCOutUcastPkts, ifHCOutMulticastPkts, ifHCOutBroadcastPkts, ifHCInOctets, ifHCOutOctets, ifInOctets, ifOutOctets, ifInUnknownProtos, ifInErrors, ifOutErrors, ifMtu, ifInUcastPkts, ifInMulticastPkts, ifInBroadcastPkts, ifInDiscards, ifOutUcastPkts, ifOutMulticastPkts, ifOutBroadcastPkts, ifOutDiscards, ifPromiscuousMode }

STATUS current

DESCRIPTION

A collection of objects providing information specific to higher speed (greater than 650,000,000 bits/second) packet-oriented network interfaces.

::= { ifGroups 6 }



**ifRcvAddressGroupOBJECT-GROUP**

OBJECTS { ifRcvAddressStatus, ifRcvAddressType }

STATUS current

DESCRIPTION

A collection of objects providing information on the multiple addresses which an interface receives.

::= { ifGroups 7 }

**ifStackGroup2OBJECT-GROUP**

OBJECTS { ifStackStatus, ifStackLastChange }

STATUS current

DESCRIPTION

A collection of objects providing information on the layering of MIB-II interfaces.

::= { ifGroups 11 }

**ifCounterDiscontinuityGroup OBJECT-GROUP**

OBJECTS { ifCounterDiscontinuityTime }

STATUS current

DESCRIPTION

A collection of objects providing information specific to interface counter discontinuities.

::= { ifGroups 13 }

**linkUpDownNotificationsGroup NOTIFICATION-GROUP**

NOTIFICATIONS { linkUp, linkDown }

STATUS current

DESCRIPTION

The notifications which indicate specific changes in the value of ifOperStatus.

::= { ifGroups 14 }

## 非推奨の定義：オブジェクト

### インターフェイス テスト テーブル

このオブジェクト グループはオプションであり、推奨されていません。ただし、メディア固有の MIB ではこのグループの実装が必須の場合があります。このテーブルは ifExtnsTestTable に代わるものです。

**ifTestTable OBJECT-TYPE**

SYNTAX SEQUENCE OF IfTestEntry

MAX-ACCESS not-accessible

STATUS deprecated

DESCRIPTION

This table contains one entry per interface. It defines objects which allow a network manager to instruct an agent to test an interface for various faults. Tests for an interface are defined in the media-specific MIB for that interface. After invoking a test, the object ifTestResult can be read to

determine the outcome. If an agent cannot perform the test, ifTestResult is set to so indicate. The object ifTestCode can be used to provide further test-specific or interface-specific (or even enterprise-specific) information concerning the outcome of the test. Only one test can be in progress on each interface at any one time. If one test is in progress when another test is invoked, the second test is rejected. Some agents may reject a test when a prior test is active on another interface.

Before starting a test, a manager-station must first obtain 'ownership' of the entry in the ifTestTable for the interface to be tested. This is accomplished with the ifTestId and ifTestStatus objects as follows:

```

try_again:
get (ifTestId, ifTestStatus)
while (ifTestStatus != notInUse)
/*
* Loop while a test is running or some other
* manager is configuring a test.
*/
short delay
get (ifTestId, ifTestStatus)
}
/*
* Is not being used right now -- let's compete
* to see who gets it.
*/
lock_value = ifTestId
if ( set(ifTestId = lock_value, ifTestStatus = inUse,
ifTestOwner = 'my-IP-address') == FAILURE)
/*
* Another manager got the ifTestEntry -- go
* try again
*/
goto try_again;
/*
* I have the lock
*/
set up any test parameters.
/*
* This starts the test
*/
set(ifTestType = test_to_run);

```

Wait for test completion by polling ifTestResult when test completes, agent sets ifTestResult agent also sets ifTestStatus = 'notInUse' retrieve any additional test results, and ifTestId if (ifTestId == lock\_value+1) results are valid.

A manager station first retrieves the value of the appropriate ifTestId and ifTestStatus objects, periodically repeating the retrieval if necessary, until the value of ifTestStatus is 'notInUse'. The manager station then tries to set the same ifTestId object to the value it just retrieved, the same ifTestStatus object to 'inUse', and the corresponding ifTestOwner object to a value indicating itself. If the set operation succeeds then the manager has obtained ownership of the ifTestEntry, and the value of the ifTestId object is incremented by the agent (per the semantics of TestAndIncr). Failure of the set operation indicates that some other manager has obtained ownership of the ifTestEntry.

Once ownership is obtained, any test parameters can be setup, and then the test is initiated by setting ifTestType. On completion of the test, the agent sets ifTestStatus to 'notInUse'. Once this occurs, the manager can retrieve the results. In the (rare) event that the invocation of tests by two network managers were to overlap, then there would be a possibility that the first test's results might be overwritten by the second test's results prior to the first results being read. This unlikely circumstance can be detected by a network manager retrieving ifTestId at the same time as retrieving the test results, and ensuring that the results are for the desired request.

If ifTestType is not set within an abnormally long period of time after ownership is obtained, the agent should time-out the manager, and reset the value of the ifTestStatus object back to 'notInUse'. It is suggested that this time-out period be 5 minutes.

In general, a management station must not retransmit a request to invoke a test for which it does not receive a response; instead, it properly inspects an agent's MIB to determine if the invocation was successful. Only if the invocation was unsuccessful, is the invocation request retransmitted.

Some tests may require the interface to be taken off-line in order to execute them, or may even require the agent to reboot after completion of the test. In these circumstances, communication with the management station invoking the test may be lost until after completion of the test. An agent is not required to support such tests. However, if such tests are supported, then the agent should make every effort to transmit a response to the request which invoked the test prior to losing communication. When the agent is restored to normal service, the results of the test are properly made available in the appropriate objects.

Note that this requires that the ifIndex value assigned to an interface must be unchanged even if the test causes a reboot. An agent must reject any test for which it cannot, perhaps due to resource constraints, make available at least the minimum amount of information after that test completes.

::= { ifMIBObjects 3 }

#### **ifTestEntry OBJECT-TYPE**

SYNTAX IfTestEntry

MAX-ACCESS not-accessible

STATUS deprecated

DESCRIPTION

An entry containing objects for invoking tests on an interface.

AUGMENTS { ifEntry }

::= { ifTestTable 1 }

IfTestEntry ::=

SEQUENCE { ifTestId TestAndIncr, ifTestStatus INTEGER, ifTestType AutonomousType, ifTestResult INTEGER, ifTestCode OBJECT IDENTIFIER, ifTestOwnerOwnerString }

**ifTestId OBJECT-TYPE**

SYNTAX TestAndIncr

MAX-ACCESS read-write

STATUS deprecated

DESCRIPTION

This object identifies the current invocation of the interface's test.

::= { ifTestEntry 1 }

**ifTestStatus OBJECT-TYPE**

SYNTAX INTEGER { notInUse(1), inUse(2) }

MAX-ACCESS read-write

STATUS deprecated

DESCRIPTION

This object indicates whether or not some manager currently has the necessary 'ownership' required to invoke a test on this interface. A write to this object is only successful when it changes its value from 'notInUse(1)' to 'inUse(2)'. After completion of a test, the agent resets the value back to 'notInUse(1)'.

::= { ifTestEntry 2 }

**ifTestType OBJECT-TYPE**

SYNTAX AutonomousType

MAX-ACCESS read-write

STATUS deprecated

DESCRIPTION

A control variable used to start and stop operator-initiated interface tests. Most OBJECT IDENTIFIER values assigned to tests are defined elsewhere, in association with specific types of interface. However, this document assigns a value for a full-duplex loopback test, and defines the special meanings of the subject identifier:

**noTest OBJECT IDENTIFIER ::= { 0 0 }**

When the value noTest is written to this object, no action is taken unless a test is in progress, in which case the test is aborted. Writing any other value to this object is only valid when no test is currently in progress, in which case the indicated test is initiated.

When read, this object always returns the most recent value that ifTestType was set to. If it has not been set since the last initialization of the network management subsystem on the agent, a value of noTest is returned.

::= { ifTestEntry 3 }

**ifTestResult OBJECT-TYPE**

SYNTAX INTEGER { none(1), -- no test yet requested success(2), inProgress(3), notSupported(4), unAbleToRun(5), -- due to state of system aborted(6), failed(7) }

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

This object contains the result of the most recently requested test, or the value none(1) if no tests have been requested since the last reset. Note that this facility provides no provision for saving the results of one test when starting another, as could be required if used by multiple managers concurrently.

::= { ifTestEntry 4 }

#### **ifTestCode OBJECT-TYPE**

SYNTAX OBJECT IDENTIFIER

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

This object contains a code which contains more specific information on the test result, for example an error-code after a failed test. Error codes and other values this object may take are specific to the type of interface and/or test. The value may have the semantics of either the AutonomousType or InstancePointer textual conventions as defined in RFC 2579. The identifier is testCodeUnknown OBJECT IDENTIFIER ::= { 0 0 } and defined for use if no additional result code is available.

::= { ifTestEntry 5 }

#### **ifTestOwner OBJECT-TYPE**

SYNTAX OwnerString

MAX-ACCESS read-write

STATUS deprecated

DESCRIPTION

The entity which currently has the 'ownership' required to invoke a test on this interface.

::= { ifTestEntry 6 }

## 非推奨の定義 : グループ

#### **ifGeneralGroup OBJECT-GROUP**

OBJECTS { ifDescr, ifType, ifSpeed, ifPhysAddress, ifAdminStatus, ifOperStatus, ifLastChange, ifLinkUpDownTrapEnable, ifConnectorPresent, ifHighSpeed, ifName }

STATUS deprecated

DESCRIPTION

A collection of objects deprecated in favour of ifGeneralInformationGroup.

::= { ifGroups 1 }

#### **ifTestGroup OBJECT-GROUP**

OBJECTS { ifTestId, ifTestStatus, ifTestType, ifTestResult, ifTestCode, ifTestOwner }

STATUS deprecated

DESCRIPTION

A collection of objects providing the ability to invoke tests on an interface.

::= { ifGroups 8 }

#### **ifStackGroup OBJECT-GROUP**

OBJECTS { ifStackStatus }

STATUS deprecated

DESCRIPTION

The previous collection of objects providing information on the layering of MIB-II interfaces.

::= { ifGroups 9 }

#### **ifOldObjectsGroupOBJECT-GROUP**

OBJECTS { ifInNUcastPkts, ifOutNUcastPkts, ifOutQLen, ifSpecific }

STATUS deprecated

DESCRIPTION

The collection of objects deprecated from the original MIB-II interfaces group.

::= { ifGroups 12 }

## 非推奨の定義：準拠性

#### **ifCompliance MODULE-COMPLIANCE**

STATUS deprecated

DESCRIPTION

A compliance statement defined in a previous version of this MIB module, for SNMP entities which have network interfaces.

MODULE -- this module

MANDATORY-GROUPS { ifGeneralGroup, ifStackGroup }

GROUP ifFixedLengthGroup

DESCRIPTION

This group is mandatory for all network interfaces which are character-oriented or transmit data in fixed-length transmission units.

GROUP ifHCFixedLengthGroup

DESCRIPTION

This group is mandatory only for those network interfaces which are character-oriented or transmit data in fixed-length transmission units, and for which the value of the corresponding instance of ifSpeed is greater than 20,000,000 bits/second.

GROUP ifPacketGroup

DESCRIPTION

This group is mandatory for all network interfaces which are packet-oriented.

GROUP ifHCPacketGroup

DESCRIPTION

This group is mandatory only for those network interfaces which are packet-oriented and for which the value of the corresponding instance of ifSpeed is greater than 650,000,000 bits/second.

GROUP ifTestGroup

DESCRIPTION

This group is optional. Media-specific MIBs which require interface tests are strongly encouraged to use this group for invoking tests and reporting results. A medium specific MIB which has mandatory tests may make implementation of this group mandatory.

GROUP ifRevAddressGroup

DESCRIPTION

The applicability of this group MUST be defined by the media-specific MIBs. Media-specific MIBs must define the exact meaning, use, and semantics of the addresses in this group.

OBJECT ifLinkUpDownTrapEnable

MIN-ACCESS read-only

DESCRIPTION

Write access is not required.

OBJECT ifPromiscuousMode

MIN-ACCESS read-only

DESCRIPTION

Write access is not required.

OBJECT ifStackStatus

SYNTAX INTEGER { active(1) } -- subset of RowStatus

MIN-ACCESS read-only

DESCRIPTION

Write access is not required, and only one of the six enumerated values for the RowStatus textual convention need be supported, specifically: active(1).

OBJECT ifAdminStatus

SYNTAX INTEGER { up(1), down(2) }

MIN-ACCESS read-only

DESCRIPTION

Write access is not required, nor is support for the value testing(3).

::= { ifCompliances 1 }

#### **ifCompliance2 MODULE-COMPLIANCE**

STATUS deprecated

DESCRIPTION

A compliance statement defined in a previous version of this MIB module, for SNMP entities which have network interfaces.

MODULE -- this module

MANDATORY-GROUPS { ifGeneralInformationGroup, ifStackGroup2,  
ifCounterDiscontinuityGroup }

GROUP ifFixedLengthGroup

DESCRIPTION

This group is mandatory for all network interfaces which are character-oriented or transmit data in fixed-length transmission units.

GROUP ifHCFixedLengthGroup

## DESCRIPTION

This group is mandatory only for those network interfaces which are character-oriented or transmit data in fixed-length transmission units, and for which the value of the corresponding instance of ifSpeed is greater than 20,000,000 bits/second.

GROUP ifPacketGroup

## DESCRIPTION

This group is mandatory for all network interfaces which are packet-oriented.

GROUP ifHCPacketGroup

## DESCRIPTION

This group is mandatory only for those network interfaces which are packet-oriented and for which the value of the corresponding instance of ifSpeed is greater than 650,000,000 bits/second.

GROUP ifRcvAddressGroup

## DESCRIPTION

The applicability of this group MUST be defined by the media-specific MIBs. Media-specific MIBs must define the exact meaning, use, and semantics of the addresses in this group.

OBJECT ifLinkUpDownTrapEnable

MIN-ACCESS read-only

## DESCRIPTION

Write access is not required.

OBJECT ifPromiscuousMode

MIN-ACCESS read-only

## DESCRIPTION

Write access is not required.

OBJECT ifStackStatus

SYNTAX INTEGER { active(1) } -- subset of RowStatus

MIN-ACCESS read-only

## DESCRIPTION

Write access is not required, and only one of the six enumerated values for the RowStatus textual convention need be supported, specifically: active(1).

OBJECT ifAdminStatus

SYNTAX INTEGER { up(1), down(2) }

MIN-ACCESS read-only

## DESCRIPTION

Write access is not required, nor is support for the value testing(3).

OBJECT ifAlias

MIN-ACCESS read-only

## DESCRIPTION

Write access is not required.

::= { ifCompliances 2 }