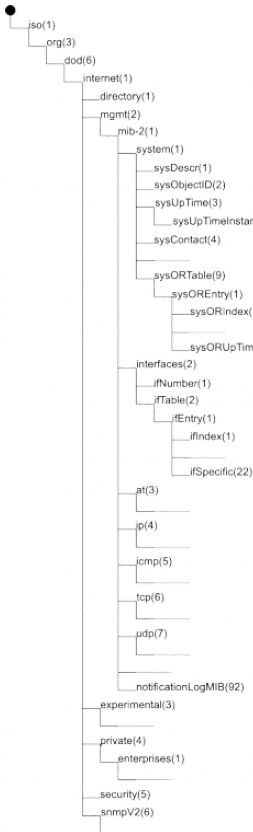


–Table des matières

- [Les MIBs](#)
 - [C'est quoi une MIB ?](#)
 - [Structure d'une MIB](#)
 - [Illustration](#)
 - [La « BRIDGE-MIB »](#)
 - [Mais encore...](#)
 - [Interro surprise 1](#)
 - [Et en mode graphique ?](#)

Les MIBs



C'est quoi une MIB ?

Il s'agit donc la base d'informations de gestion. Il y a, nous commençons à bien le comprendre :

- des informations à consulter,
- des paramètres à modifier,
- des alarmes à émettre...

Tout ceci, en principe, de façon indépendante du matériel et du logiciel. Il faut donc que SNMP permette de retrouver ces informations et d'agir sur les paramètres de façon indépendante du matériel, comme du logiciel.

Une MIB se présente comme une base de données normalisée, qui permettra de lire et d'écrire sur les équipements distants, de façon également normalisée. Ce sera à l'agent lui-même de faire l'interface entre les informations récupérables sur la plateforme où il est installé et le jargon utilisé par SNMP.

Structure d'une MIB

Elle est extrêmement simple pour un informaticien, et donc assez compliquée pour un cerveau humain « normal ».

Elle est organisée hiérarchiquement, de la même façon que l'arborescence des domaines Internet.

Elle peut contenir des scalaires (valeurs uniques) ou des tableaux de scalaires.

Non seulement la structure est normalisée, mais également les appellations des diverses rubriques. Ces appellations ne servent à rien, si ce n'est à rendre les choses plus lisibles (ce qui peut éventuellement prêter à rire). En réalité, chaque niveau de la hiérarchie est repéré par un index numérique et SNMP n'utilise que cette façon de faire. Une MIB est une branche, qui vient se greffer sur l'arbre général. Nous en avons juste à côté une vue très simplifiée, qu'il nous sera possible de contempler avec plus de détails par la suite avec l'outil `snmptranslate`.

Tout constructeur d'un matériel spécifique peut développer une MIB pour ce matériel, cette MIB devra prendre sa place dans l'arbre, sans piétiner celle des voisins, et c'est le rôle de l'IETF ou de l'IANA d'attribuer un point de branchement pour le matériel de ce constructeur.

Une MIB n'étant finalement qu'une branche, elle peut nécessiter l'usage de d'autres MIBS pour se rattacher à l'arbre. Prenons un exemple. HP peut développer une MIB générique à tous ses matériels qui viendrait se greffer sur la branche `private.enterprise`, puis des MIBs spécifiques à chaque matériel qui viendraient elles-mêmes se développer dans la MIB générique de HP.

Illustration

Sur l'arborescence en illustration, nous constatons que le sommet est représenté par un point originel, puisque chaque embranchement dispose d'un nom (iso, org, internet...) et aussi d'un numéro (1, 3, 6, 1...).

Intéressons-nous au chemin qui mène à `sysDescr` soit :

- `.iso.org.dod.internet.mgmt.mib-2.system.sysDescr` en mode texte ;
- `.1.3.6.1.2.1.1.1` en mode numérique.

Reprenons notre switch et interrogeons-le de diverses manières, à propos de ce chemin :

```
~$ snmpwalk -of -c public -v 1 172.16.252.2 .1.3.6.1.2.1.1.1
```

- `snmpwalk` signifie que l'on désire parcourir un arbre snmp ;
- `-of` signifie que l'on désire afficher l'arborescence « full-text » ;
- `-c public` signifie que l'on se place dans la communauté « public », ceci prendra tout son sens un peu plus tard ;
- `-v 1` signifie que l'on utilise la version 1 du protocole snmp ;
- vient ensuite l'adresse IP de l'agent que l'on désire interroger ;
- vient enfin, en mode numérique, le point d'embranchement à partir duquel nous désirons parcourir l'arbre.

Nous obtenons la réponse :

```
.iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0 = STRING: ProCurve J4899B Switch 2650, revision H.10.35, ROM H.08.02 (/sw/code/build/fish(mkfs))
```

Une chaîne de caractères qui identifie le type de matériel (system descriptor). Le chemin qui mène à cette information est noté en mode « full-text », comme il a été demandé.

L'outil `snmpwalk` s'avère d'une grande puissance, pour peu qu'on arrive à le dompter un tant soit peu. Réutilisons-le en remontant d'un niveau dans le point de départ :

```
~$ snmpwalk -of -c public -v 1 172.16.252.2 .1.3.6.1.2.1.1
```

(nous avons enlevé le dernier .1 dans la définition du point de départ) :

```
.iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0 = STRING: ProCurve J4899B Switch 2650, revision H.10.35, ROM H.08.02 (/sw/code/build/fish(mkfs))
.iso.org.dod.internet.mgmt.mib-2.system.sysObjectID.0 = OID: .iso.org.dod.internet.private.enterprises.11.2.3.7.11.44
.iso.org.dod.internet.mgmt.mib-2.system.sysUpTime.sysUpTimeInstance = Timeticks: (1071094654) 123 days, 23:15:46.54
.iso.org.dod.internet.mgmt.mib-2.system.sysContact.0 = STRING: sysop
.iso.org.dod.internet.mgmt.mib-2.system.sysName.0 = STRING: ProCurve Switch 2650-1
.iso.org.dod.internet.mgmt.mib-2.system.sysLocation.0 = STRING:
.iso.org.dod.internet.mgmt.mib-2.system.sysServices.0 = INTEGER: 74
```

Notez que la réponse aligne tout ce que l'agent sait dire à partir de .iso.org.dod.internet.mgmt.mib-2.system.

Bien entendu, une interrogation de la forme :

```
snmpwalk -of -c public -v 1 172.16.252.2 .iso.org.dod.internet.mgmt.mib-2.system
```

fait parfaitement le travail. Mais nous aurions pu tout aussi bien poser la question de façon beaucoup plus compacte :

```
~$ snmpwalk -of -c public -v 1 172.16.252.2 system
```

```
.iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0 = STRING: ProCurve J4899B Switch 2650, revision H.10.35, ROM H.08.02 (/sw/code/build/fish(mkfs))
.iso.org.dod.internet.mgmt.mib-2.system.sysObjectID.0 = OID: .iso.org.dod.internet.private.enterprises.11.2.3.7.11.44
.iso.org.dod.internet.mgmt.mib-2.system.sysUpTime.sysUpTimeInstance = Timeticks: (1071132769) 123 days, 23:22:07.69
.iso.org.dod.internet.mgmt.mib-2.system.sysContact.0 = STRING: sysop
.iso.org.dod.internet.mgmt.mib-2.system.sysName.0 = STRING: ProCurve Switch 2650-1
.iso.org.dod.internet.mgmt.mib-2.system.sysLocation.0 = STRING:
.iso.org.dod.internet.mgmt.mib-2.system.sysServices.0 = INTEGER: 74
```

Nous donnons ici le chemin non plus absolu, mais relatif au nœud system (pas de point initial), et le système s'y retrouve. Ceci sous-entend que chaque nœud doit disposer d'un nom unique dans l'arbre tout entier.

D'autres formats de sortie sont possibles :

- -on affiche le résultat avec l'identifiant en mode numérique ;
- -os « print MIB module-id plus last element » :

```
~$ snmpwalk -os -c public -v 1 172.16.252.2 sysName
SNMPV2-MIB::sysName.0 = STRING: ProCurve Switch 2650-1
```

Ce qui laisse imaginer que si l'on pose la question :

```
~$ snmpwalk -on -c public -v 1 172.16.252.2 SNMPV2-MIB::sysDescr
```

la réponse devrait être favorable et de la forme :

```
.1.3.6.1.2.1.1.1.0 = STRING: ProCurve J4899B Switch 2650, revision H.10.35, ROM H.08.02 (/sw/code/build/fish(mkfs))
```

ce qui est bien le cas.

La « BRIDGE-MIB »

Un switch est tout de même un objet un peu particulier et ce que nous avons vu jusqu'ici va atteindre ses limites.

Ainsi, au bout de la branche .1.3.6.1.2.1.17.1.2, nous devons normalement trouver le nombre de ports dont dispose le switch (il faut être un initié pour le savoir). Le bout de la branche devant obligatoirement se terminer par un 0, nous l'avons remarqué lors des démonstrations précédentes, posons la question avec cette fois-ci un snmpget, qui ne sait extraire qu'une seule information, et qui doit donc être obligatoirement posée en indiquant le bout de la branche :

```
$ snmpget -of -c public -v 2c 172.16.252.2 .1.3.6.1.2.1.17.1.2.0
.iso.org.dod.internet.mgmt.mib-2.17.1.2.0 = INTEGER: 50
```

Il y a bien 50 ports sur notre switch HP :

- 48 ports 10/100 Mb/s ;
- 2 ports 10/100/1000 Mb/s.

Mais cette réponse apporte, en prime, une bonne et une mauvaise nouvelle.

La bonne nouvelle, c'est que notre switch comprend bien la question, puisqu'il donne une réponse.

La mauvaise, c'est que notre commande snmpget ne sait pas traduire la totalité du chemin en mode « full-text », comme il a été pourtant demandé (option -of). En effet, sa réponse est :

```
.iso.org.dod.internet.mgmt.mib-2.17.1.2.0 = INTEGER: 50
```

autrement dit, il manque des pages dans le dictionnaire, snmpget sait traduire dans la suite .1.3.6.1.2.1.17.1.2.0 la seule partie .1.3.6.1.2.1 et pas la suite 17.1.2, c'est pourquoi cette suite reste numérique dans la réponse. La bonne nouvelle dans la mauvaise nouvelle, c'est qu'il reste possible d'interroger un agent SNMP sur une information, même si l'on ne dispose pas des pages manquantes relatives à la définition de cette information, du moment que l'on connaît le chemin numérique pour y accéder.

Nous allons tout de même demander à notre snmpget d'intégrer ces pages qui manquent (cette MIB¹), que nous avons récupérée grâce au paquet snmp-mibs-downloader que nous avons installé avec un aptitude install des familles et qui nous a mis tout ça dans /usr/share/mibs :

```
snmpget -m +usr/share/mibs/ietf/BRIDGE-MIB -of -c public -v 2c 172.16.252.2 .1.3.6.1.2.1.17.1.2.0
.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dBase.dot1dBaseNumPorts.0 = INTEGER: 50 ports
```

Voici une réponse nettement plus compréhensible, la suite .1.3.6.1.2.1.17.1.2.0 est cette fois-ci complètement interprétée...

Inversement, nous pouvons opérer ainsi :

```
snmpget -m /usr/share/mibs/ietf/BRIDGE-MIB -on -c public -v 2c 172.16.252.2 .iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dBase.dot1dBaseNumPorts.0
.1.3.6.1.2.1.17.1.2.0 = INTEGER: 50 ports
```

Nous avons posé la question en indiquant le chemin en mode texte en demandant la réponse au format numérique. Mais aurions-nous pu faire cela sans l'aide de la BRIDGE-MIB ?

```
$ snmpget -on -c public -v 2c 172.16.252.2 .iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dBase.dot1dBaseNumPorts.0
.iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dBase.dot1dBaseNumPorts.0: Unknown Object Identifier (Sub-id not found: mib-2 -> dot1dBridge)
```

Bien sûr que non, sans son aide, `snmpget` ne sait pas traduire le chemin exprimé en langage « humain ».

Dernière vérification pour la route, nous posons la même question, cette fois-ci à un switch de marque D-LINK :

```
$ snmpget -Of -c public -v 2c 172.16.252.4 .1.3.6.1.2.1.1.1.0
.iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0 = STRING: DGS-3024 Gigabit Ethernet Switch
```

il s'agit d'un switch DGS-3024, modèle notoirement proposé par D-LINK.

```
$ snmpget -m +usr/share/mibs/ietf/BRIDGE-MIB -On -c public -v 2c 172.16.252.4 .iso.org.dod.internet.mgmt.mib-2.dot1dBridge.dot1dBase.dot1dBaseNumPorts.0
.1.3.6.1.2.1.17.1.2.0 = INTEGER: 24 ports
```

En effet, ce modèle propose 24 ports 10/100/1000 Mb/s.

Ce petit exemple était destiné à démontrer que deux switches de marque bien différente savent répondre à la même question, si l'information demandée est définie dans la BRIDGE-MIB.

Bien sûr chaque constructeur va pouvoir apporter à ses équipements des particularités supplémentaires et spécifiques à sa marque et à son modèle. Elles sont dans ce cas placées après le nœud `enterprises`. Nous pourrions toujours obtenir ces données, mais sans les MIBs associées, il sera difficile, voir impossible de déterminer ce que ces données représentent.

Encore une fois, pourquoi se passer d'un peu d'humour ?

```
~$ snmpwalk -Of -c public -v 1 172.16.252.2 enterprises
```

va nous donner :

```
.iso.org.dod.internet.private.enterprises.9.9.23.1.1.1.1.2.1 = INTEGER: 1
.iso.org.dod.internet.private.enterprises.9.9.23.1.1.1.1.2.2 = INTEGER: 1
...
.iso.org.dod.internet.private.enterprises.9.9.23.1.1.1.1.2.50 = INTEGER: 1
.iso.org.dod.internet.private.enterprises.9.9.23.1.2.1.1.3.50.1 = INTEGER: 1
.iso.org.dod.internet.private.enterprises.9.9.23.1.2.1.1.4.50.1 = Hex-STRING: AC 10 FC 03
.iso.org.dod.internet.private.enterprises.9.9.23.1.2.1.1.5.50.1 = STRING: "ProCurve J4899B Switch 2650, revision H.10.35, ROM H.08.02 (/sw/code/build/fish(mkfs))"
.iso.org.dod.internet.private.enterprises.9.9.23.1.2.1.1.6.50.1 = Hex-STRING: 00 17 08 E4 C5 00
.iso.org.dod.internet.private.enterprises.9.9.23.1.2.1.1.7.50.1 = STRING: "49"
.iso.org.dod.internet.private.enterprises.9.9.23.1.2.1.1.8.50.1 = STRING: "ProCurve J4899B Switch 2650, revision H.10.35, ROM H.08.02 (/sw/code/build/fish(mkfs))"
.iso.org.dod.internet.private.enterprises.9.9.23.1.2.1.1.9.50.1 = STRING: "8"
.iso.org.dod.internet.private.enterprises.9.9.23.1.3.1.0 = INTEGER: 1
.iso.org.dod.internet.private.enterprises.9.9.23.1.3.2.0 = INTEGER: 0
.iso.org.dod.internet.private.enterprises.9.9.23.1.3.3.0 = INTEGER: 0
...
```

Il y en a pour 7346 lignes sur notre Procurve. Le D-Link DGS-3024 quant à lui n'en sortira que 2205 ...

Sans les MIBs qui vont avec, il sera quasiment impossible de savoir à quoi correspondent ces informations, car les MIBs, en plus de traduire les chemins numériques en texte, peuvent donner des indications supplémentaires sur la nature de l'information extraite.

Mais encore...

Les MIBs ne se contentent pas de fournir une traduction entre les modes numériques et textuels. Elles fournissent aussi quelques explications sur les paramètres qu'elles prennent en charge. La commande `snmptranslate` permet, une fois domestiquée, d'obtenir quelque documentation supplémentaire :

```
~$ snmptranslate -Tp SNMPv2-MIB::system
+--system(1)
|
|--R-- String sysDescr(1)
|   Textual Convention: DisplayString
|   Size: 0..255
+--R-- ObjID sysObjectID(2)
+--R-- TimeTicks sysUpTime(3)
|   |
|   +--sysUpTimeInstance(0)
|   |
|--RW- String sysContact(4)
|   Textual Convention: DisplayString
|   Size: 0..255
+--RW- String sysName(5)
|   Textual Convention: DisplayString
|   Size: 0..255
+--RW- String sysLocation(6)
|   Textual Convention: DisplayString
|   Size: 0..255
+--R-- INTEGER sysServices(7)
|   Range: 0..127
+--R-- TimeTicks sysORLastChange(8)
|   Textual Convention: TimeStamp
|
+--sysORTable(9)
|
+--sysOREntry(1)
|   Index: sysORIndex
|   |
|   +-- ---- INTEGER sysORIndex(1)
|   |   Range: 1..2147483647
|   +--R-- ObjID sysORID(2)
|   +--R-- String sysORDescr(3)
|   |   Textual Convention: DisplayString
|   |   Size: 0..255
|   +--R-- TimeTicks sysORUpTime(4)
|   |   Textual Convention: TimeStamp
```

Nous obtenons, en art ASCII, une vue arborescente de la branche `system`.

Un peu de traduction pure et simple :

```
~$ snmptranslate -On SNMPv2-MIB::sysDescr
.1.3.6.1.2.1.1.1
```

```
~$ snmptranslate -Ot .1.3.6.1.2.1.1.1
```

```
SNMPv2-MIB::sysDescr
```

```
~$ snmptranslate -Of .1.3.6.1.2.1.1.1  
.iso.org.dod.internet.mgmt.mib-2.system.sysDescr
```

Et éventuellement quelques explications :

```
~$ snmptranslate -Ot -Td .1.3.6.1.2.1.1.1  
SNMPv2-MIB::sysDescr  
sysDescr OBJECT-TYPE  
-- FROM SNMPv2-MIB, RFC1213-MIB  
-- TEXTUAL CONVENTION DisplayString  
SYNTAX OCTET STRING (0..255)  
DISPLAY-HINT "255a"  
MAX-ACCESS read-only  
STATUS current  
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."  
::= { iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) system(1) 1 }
```

Normalement, chaque bout de branche dispose d'une description. Ne vous attendez jamais à voir apparaître autre chose que de l'anglais, bien entendu.

```
~$ snmptranslate -Ot -Td SNMPv2-MIB::sysORIndex  
SNMPv2-MIB::sysORIndex  
sysORIndex OBJECT-TYPE  
-- FROM SNMPv2-MIB  
SYNTAX INTEGER (1..2147483647)  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION "The auxiliary variable used for identifying instances  
of the columnar objects in the sysORTable."  
::= { iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) system(1) sysORTable(9) sysOREntry(1) 1 }
```

Notez également que ceci peut parfois tourner au jeu de piste.

Interro surprise 1

Est-ce que ceci aurait des chances de fonctionner ?

```
~$ snmptranslate -Ot -Td .1.3.6.1.2.1.17.1.2.0
```

Et ceci :

```
~$ snmptranslate -Ot -Td -m /usr/share/mibs/ietf/BRIDGE-MIB .1.3.6.1.2.1.17.1.2.0
```

Vous trouverez les réponses à la fin de l'exposé.

Et en mode graphique ?

Il existe un outil, `TKmib` qui, comme son nom l'indique, utilise TK et n'est donc pas des plus esthétiques. Ce genre d'outil, nommé « browser de MIBs » permet de faire de façon graphique, ce que l'on a vu avec `nmpwalk`, `snmpget` et `snmptranslate`.

Ce n'est pas très joli, mais ça fait le travail.

[1\)](#) appelée BRIDGE-MIB et définie dans la [rfc 1493](#)



[DOKUMIKI](#) [RSS](#) [XML FEED](#)

Dernière modification:: le 26/12/2010 à 11:20

Cette création est mise à disposition sous un [contrat Creative Commons](#).

