

Compilateurs MIB et chargement des MIB

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Conventions](#)

[Problèmes courants de chargement de la base MIB](#)

[Commande de charge](#)

[Non-concordance sur les définitions de type de données](#)

[Redéfinitions de l'identificateur d'objet](#)

[Définitions des types de données intégrés](#)

[Dimensions alternatives](#)

[Identificateurs d'objet impairs](#)

[Définitions de déroutement](#)

[Compilateurs basés sur RFC 14xx contre Compilateurs basés sur RFC 19xx](#)

[Chargement et compilation de MIBS dans un NMS tiers](#)

[À partir de l'interface utilisateur graphique de HP OpenView ou d'IBM NetView](#)

[À partir de l'interface de ligne de commande de HP OpenView ou IBM NetView](#)

[Informations connexes](#)

[Introduction](#)

La plupart des systèmes d'administration de réseaux (NMS) fournissent une manière pour que l'utilisateur charge les MIB. Le chargement d'un MIB est une manière dont les NMS peuvent se renseigner sur de nouveaux objets MIB, tels que leurs noms, identifiants d'objet (OID), et le genre de type de données (par exemple, compteur).

La base MIB peut être analysée lors du chargement ou plus tard, par exemple, lorsqu'une application NMS s'exécute. Le logiciel qui effectue l'analyse est un compilateur MIB.

Toute MIB syntaxiquement correcte doit être analysée avec succès par le compilateur MIB de n'importe quel fournisseur. Malheureusement, différents compilateurs MIB peuvent présenter des bizarreries différentes.

Cisco s'efforce en permanence de s'assurer que les MIB publiées aux clients sont correctes sur le plan syntaxique. Cisco évite également les constructions MIB qui se sont avérées problématiques dans les produits NMS courants. Malgré ces efforts, il n'est pas possible de satisfaire les bizarreries dans tous les compilateurs MIB sur le terrain.

Ce document aborde certains des problèmes courants et suggère des solutions. Si vous rencontrez l'un de ces problèmes avec le compilateur MIB de votre fournisseur (à l'exception du

problème [RFC 14xx contre RFC 19xx](#)), il est dû à une erreur dans ce compilateur MIB. Vous pouvez demander à votre ou vos fournisseurs de corriger leurs compilateurs.

Conditions préalables

Conditions requises

Les lecteurs de ce document doivent être familiarisés avec les MIB.

Components Used

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

For more information on document conventions, refer to the [Cisco Technical Tips Conventions](#).

Problèmes courants de chargement de la base MIB

Commande de charge

L'ordre de chargement est le problème le plus important et le plus courant lorsque vous chargez des MIB. De nombreuses MIB utilisent des définitions définies dans d'autres MIB. Ces définitions sont répertoriées dans la clause `IMPORTS` située en haut de la MIB.

Par exemple, si MIB **mumble** importe une définition à partir de la **bosse** MIB, certains compilateurs MIB vous demandent de charger la **bosse** MIB avant de charger la **bogue** MIB. Si l'ordre de chargement est incorrect, le compilateur prétendra que les MIB importées ne sont pas définies.

Il s'agit d'une liste de MIB à partir de laquelle de nombreuses autres MIB importent et de l'ordre dans lequel vous devez les charger. Cela prend probablement en charge 95 % des problèmes d'ordre de charge (la plupart des autres MIB peuvent être chargés dans n'importe quel ordre) :

- SNMPv2-SMI.my
- SNMPv2-TC.my
- SNMPv2-MIB.my
- RFC1213-MIB.my
- IF-MIB.my
- CISCO-SMI.my
- CISCO-PRODUCTS-MIB.my
- CISCO-TC.my

Remarque : Si vous chargez les versions v1 de ces MIB, le nom du fichier MIB ressemblera en fait à IF-MIB-V1SMI.my ("-V1SMI " est ajouté au nom des MIB qui ont été convertis de v2 à v1). L'exception à cela est la MIB [RFC1213-MIB.my](#), qui n'existe qu'en version v1 (c'est-à-dire qu'il n'y

a pas de RFC1213-MIB-V1SMI.my).

Si vous tentez de charger une autre MIB et si le compilateur se plaint d'éléments non définis, identifiez les MIB que cette MIB importe et vérifiez que vous avez chargé toutes les autres MIB en premier.

Remarque : Pour chaque MIB, vous pouvez voir la liste exacte des MIB qui doivent être chargées avant elle, avec l'ordre de compilation exact, dans [SNMP Object Navigator > View & Download MIBs](#) ; sélectionnez **Afficher les dépendances MIB et téléchargez MIB**.

[Non-concordance sur les définitions de type de données](#)

Bien que les définitions de type de données MIB de Cisco ne soient pas incompatibles, vous pouvez trouver que c'est le cas pour certaines MIB RFC standard. Exemple :

- MIB **mumble** définit : `Type de données some ::= INTEGER(0.100)`
- La **tumeur** MIB définit : `CertainsType de données ::= INTEGER(1.50)`

Cet exemple est considéré comme une erreur insignifiante et la MIB se charge correctement avec un message d'avertissement.

L'exemple suivant est considéré comme une erreur non négligeable (même si les deux définitions sont essentiellement équivalentes) et la MIB n'est pas analysée correctement.

- MIB **mumble** définit : `Type de données some ::= chaîneAffichage`
- La **tumeur** MIB définit : `CertainDatatype ::= CHAÎNE OCTET (TAILLE(0.255))`

Si votre compilateur MIB traite ces erreurs comme des erreurs, ou si vous souhaitez supprimer les messages d'avertissement, modifiez l'une des MIB qui définissent ce même type de données afin que les définitions correspondent.

[Redéfinitions de l'identificateur d'objet](#)

Vous pouvez rencontrer des redéfinitions OID si vous chargez ces MIB (bien que cette erreur puisse se produire dans d'autres cas) :

- [ANCIEN CISCO-CPU-MIB.my](#)
- [ANCIEN CISCO-ENV-MIB.my](#)
- [ANCIEN CISCO-MEMORY-MIB.my](#)
- [ANCIEN CISCO-SYSTEM-MIB.my](#)

Exemple :

- OLD-CISCO-CPU-MIB.my définit : `IDENTIFICATEUR D'OBJET LCpu ::= { local 1 }`
- OLD-CISCO-ENV-MIB.my définit : `IDENTIFICATEUR D'OBJET lenv ::= { local 1 }`

Lorsque vous chargez ces deux MIB, le compilateur MIB peut se plaindre que l'`IDENTIFIANT OBJET lcpu` est redéfini avec un nouveau nom `lenv`. Les anciens-CISCO-MEMORY-MIB.my et OLD-CISCO-SYSTEM-MIB.my donnent de nouveaux noms à `{ local 1 }`.

Cette erreur est traitée comme une erreur insignifiante et la MIB se charge correctement avec un message d'avertissement.

Si la MIB ne se charge pas correctement ou si vous souhaitez supprimer le message

d'avertissement, modifiez l'une des MIB de sorte que toutes les MIB utilisent le même nom.

Définitions des types de données intégrés

De nombreux compilateurs MIB possèdent une connaissance intégrée de certains types de données, tels que DisplayString. Certains de ces compilateurs se plaignent de voir une définition de ces types de données dans une MIB. Par exemple, DisplayString est défini dans SNMPv2-TC.

La solution de contournement consiste à supprimer ou commenter la définition d'infraction dans le fichier MIB.

Dimensions alternatives

Il s'agit d'un exemple syntaxiquement valide, qui indique qu'une valeur de type `MyDatatype` aura une longueur de 0, 5 ou 20 octets :

```
MyDatatype ::= OCTET STRING (SIZE(0 | 5 | 20))
```

Certains compilateurs MIB n'acceptent pas cette syntaxe. Généralement, une solution de contournement suffisante consiste à choisir l'une des tailles et à enlever les autres. Vous devriez garder la plus grande taille. Par exemple, l'exemple précédent est modifié comme suit :

```
MyDatatype ::= OCTET STRING (SIZE(20))
```

Identificateurs d'objet impairs

Certains OID sont considérés comme impairs car ils ne font pas référence à un noeud dans l'interface SMI (comme la plupart des IDENTIFICATEURS OBJET). Cependant, elles sont syntaxiquement valides. Un exemple courant est l'identificateur d'objet null, par exemple `{ 0 }`. Certains compilateurs MIB ne se soucient pas des IDENTIFICATEURS OBJET qui ne correspondent pas à un noeud dans le SMI. Voici des exemples de syntaxe MIB qui pourraient causer des problèmes à ces compilateurs :

```
zeroDotZero OBJECT IDENTIFIER ::= { 0 0 }  
myMIBObject OBJECT-TYPE  
DEFVAL { {0 0} }
```

La solution de contournement consiste à supprimer ou commenter ces types de références dans le fichier MIB.

Définitions de déroutement

Dans les MIB SNMPv1, les déroutements sont définis à l'aide de la macro `TRAP-TYPE`. Dans les MIB SNMPv2, les déroutements sont définis à l'aide de la macro `NOTIFICATION-TYPE`.

Certains compilateurs MIB n'aiment pas ces définitions dans les fichiers MIB qu'ils analysent (ils ne prennent pas en charge ces macros).

Si c'est le cas, vous pouvez supprimer les définitions de déroutement ou les commenter (par exemple, placez le délimiteur de commentaires MIB – au début des lignes).

[Compilateurs basés sur RFC 14xx contre Compilateurs basés sur RFC 19xx](#)

Les RFC 1442 à 1452 définissent le protocole SNMPv2 basé sur les parties. Ces RFC sont obsolètes par les RFC 1902 à 1908, plus récents.

En ce qui concerne la syntaxe MIB, il y a très peu de différences entre ces deux versions de SNMPv2 ; il y en a cependant. Les MIB Cisco sont actuellement basés sur les règles RFC 19xx.

Remarque : il y a quelques années, lorsque les MIB Cisco étaient basées sur RFC 14xx, certains compilateurs basés sur RFC 19xx se plaignaient de la ligne `Unsigned32 ::= TEXTUAL-CONVENTION` dans CISCO-TC.my et PNNI-MIB.my MIBs. En effet, Unsigned32 est un type de données prédéfini dans RFC 19xx. Pour cette raison, Cisco utilisait d'autres versions de ces MIB (CISCO-TC-NO-U32.my et PNNI-MIB-NO-U32.my) sans définition pour Unsigned32, pour charger dans les compilateurs qui connaissent déjà ce type de données. Cela ne s'applique plus.

[Chargement et compilation de MIBS dans un NMS tiers](#)

La meilleure et la plus efficace façon de charger les MIB, les pièges et les icônes Cisco dans les NMS tiers est d'utiliser CiscoWorks Integration Utility (Integration Utility), qui est disponible dans le cadre de CiscoWorks Common Services (ou autonome à partir de <http://www.cisco.com/cgi-bin/tablebuild.pl/cw2000-utility>), avec la carte d'utilitaire d'intégration correspondante de <http://www.cisco.com/tacpage/sw-center/cw2000/cmc3rd.shtml> et la dernière NMIDB (Network Management Integration Data Bundle). Pour plus d'informations, consultez la documentation de l'utilitaire d'intégration.

Vous pouvez également consulter la documentation de NMS tiers sur le chargement et la compilation de MIB. Ce document inclut des instructions pour HP OpenView et IBM NetView ; mais vous devez toujours consulter la documentation HP ou IBM, car les produits peuvent changer.

[À partir de l'interface utilisateur graphique de HP OpenView ou d'IBM NetView](#)

Procédez comme suit pour charger les MIB Cisco que vous souhaitez :

1. Copiez les fichiers dans le répertoire `/usr/OV/snmp_mibs` de la station de gestion du réseau. Il s'agit du répertoire par défaut dans lequel HP OpenView et IBM NetView recherchent des documents MIB. Si vous les placez ailleurs, spécifiez les noms de chemin explicites dans l'interface graphique loadmib.
2. Définissez les autorisations de sorte que vous ayez accès en lecture aux MIB.
3. Dans le menu GUI, sélectionnez **Options > Charger/Décharger les MIB**.
4. Suivez les instructions de la documentation de la plate-forme pour compiler ou charger les MIB Cisco.

[À partir de l'interface de ligne de commande de HP OpenView ou IBM NetView](#)

Émettez la commande `/opt/OV/bin/xnmloadmib -load filename`, pour charger le fichier MIB.

[Informations connexes](#)

- [Support et documentation techniques - Cisco Systems](#)