

Compiladores MIB y carga de MIB

Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Problemas comunes de carga de MIB](#)

[Pedido de carga](#)

[Discordancias en las definiciones de tipos de datos](#)

[Redefiniciones del identificador de objeto](#)

[Definiciones de tipos de datos integrados](#)

[Tamaños alternativos](#)

[Identificadores de objetos impares](#)

[Definiciones de notificaciones de trampa](#)

[Compiladores basado en RFC 14XX contra los compiladores basado en RFC 19XX](#)

[MIB que carga y de compilación en el NMS de tercera persona](#)

[Del GUI del HP OpenView o del IBM NetView](#)

[De la interfaz de línea de comando del HP OpenView o del IBM NetView](#)

[Información Relacionada](#)

[Introducción](#)

La mayoría de los sistemas de administración de red (NMS) proporcionan un método para que el usuario cargue los MIB. Cargar un MIB es una manera que un NMS aprenda sobre los nuevos objetos MIB, por ejemplo sus nombres, identificadores de objeto (OID) y la clase de tipo de datos (por ejemplo, Contador).

El MIB pudo ser analizado cuando se carga o puede ser que suceda más adelante, por ejemplo, cuando una aplicación NMS se ejecuta. El software que realiza el análisis de datos es un compilador MIB.

Cualquier MIB sintácticamente correcto se debe analizar con éxito por el compilador MIB de cualquier vendedor. Desafortunadamente, diversos compiladores MIB pueden exhibir diversos caprichos.

Cisco hace esfuerzos continuos para asegurarse de que el MIB publicado a los clientes está sintácticamente correcto. Cisco también evita las construcciones MIB que han demostrado ser problemáticas en productos NMS populares. A pesar de estos esfuerzos, no es posible satisfacer los rasgos en todos los compiladores MIB en el campo.

Este documento aborda algunos de los problemas comunes y sugiere las soluciones alternativas.

Si usted encuentra ninguno de estos problemas con el compilador MIB de su vendedor (a excepción del [RFC 14xx contra el](#) problema del [RFC 19xx](#)), es debido a una deficiencia en ese compilador MIB. Usted puede desear impulsar su vendedor o vendedores reparar sus compiladores.

prerrequisitos

Requisitos

Los Quien lea este documento deben ser familiares con el MIB.

Componentes Utilizados

Este documento no tiene restricciones específicas en cuanto a versiones de software y de hardware.

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando, asegúrese de haber comprendido el impacto que puede tener cualquier comando.

Convenciones

Para obtener más información sobre las convenciones del documento, consulte las [Convenciones de Consejos Técnicos de Cisco](#).

Problemas comunes de carga de MIB

Pedido de carga

El orden de carga es el más importante y problema común cuando usted está cargando el MIB. Mucho MIB utiliza las definiciones que se definen en el otro MIB. Estas definiciones se detallan en la cláusula IMPORTACIONES en la parte superior del MIB.

Por ejemplo, si el MIB mumble importa una definición del MIB bumble, algunos compiladores de MIB necesitan que cargue el MIB bumble antes de cargar el MIB mumble. Si usted consigue la orden de la carga incorrecta, el compilador demandará que el MIB importado es indefinido.

Ésta es una lista de MIB de las cuales importación de mucho la otra MIB, y de la orden en la cual usted debe cargarla. Esto toma probablemente el cuidado del 95 por ciento de los problemas de la orden de la carga (la mayor parte del otro MIB se puede cargar en cualquier orden):

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

- CISCO-PRODUCTS-MIB.my
- CISCO-TC.my

Note: Si usted está cargando las versiones del v1 de este MIB, el nombre de archivo MIB parecerá realmente **IF-MIB-V1SMI.my** (“-V1SMI” se agrega al nombre del MIB que se ha convertido del v2 al v1). La excepción a esto es el [RFC1213-MIB.my](#) MIB, que existe solamente como versión del v1 (es decir, no hay RFC1213-MIB-V1SMI.my).

Si usted intenta cargar otro MIB, y si el compilador se queja por los elementos indefinidos, después identifique de qué MIB está importando este MIB y verifique que usted ha cargado el resto de MIB primero.

Note: Para cada MIB, usted puede ver la lista exacta del MIB que necesita ser cargado antes de ella — con la orden exacta de la compilación — en el [SNMP Object Navigator > el MIB de la visión y de la descarga](#); seleccione las **dependencias de la visión MIB y descargue el MIB**.

[Discordancias en las definiciones de tipos de datos](#)

Aunque las definiciones de tipos de datos MIB no serán desiguales, puede ser que ese sea el caso para algunas MIB RFC estándares. Por ejemplo:

- MIB mumble define: `SomeDatatype:: = INTEGER(0..100)`
- El MIB **manosea** define: `SomeDatatype:: = INTEGER(1..50)`

A este ejemplo se lo considera como un error trivial y la MIB se carga satisfactoriamente con un mensaje de advertencia.

El siguiente ejemplo se considera un error no trivial (incluso cuando las dos definiciones son esencialmente equivalentes) y el MIB no se analiza exitosamente.

- MIB mumble define: `SomeDatatype:: = DisplayString`
- El MIB **manosea** define: `SomeDatatype:: = CADENA DE OCTETOS (SIZE(0..255))`

Si su compilador MIB trata éstos como errores, o si usted desea librarse de los mensajes de advertencia, después edite uno del MIB que define este mismo datatype de modo que las definiciones hagan juego.

[Redefiniciones del identificador de objeto](#)

Usted puede encontrar las redefiniciones OID si usted carga este MIB (aunque puede haber otros casos donde ocurre este error):

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

Por ejemplo:

- OLD-CISCO-CPU-MIB.my define: `IDENTIFICADOR del lcpuOBJECT:: = {local 1}`
- OLD-CISCO-ENV-MIB.my define: `IDENTIFICADOR del lenvOBJECT:: = {local 1}`

Cuando usted carga este dos MIB, el compilador MIB puede quejarse por el `IDENTIFICADOR DE OBJETO del lcpu` que es redefinido con un `lenv` del nuevo nombre. Los OLD-CISCO-MEMORY-MIB.my y los OLD-CISCO-SYSTEM-MIB.my dan semejantemente los nuevos nombres al `{local`

1}).

Esto se trata como un error trivial y la MIB se carga satisfactoriamente con un mensaje de advertencia.

Si el MIB no carga con éxito, o si usted desea librarse del mensaje de advertencia, después edite uno del MIB de modo que todo el MIB utilice el mismo nombre.

[Definiciones de tipos de datos integrados](#)

Muchos compiladores MIB tienen conocimiento incorporado de algunos datatypes, tales como DisplayString. Algunos de estos compiladores se quejan de si ven una definición para estos datatypes en un MIB. Por ejemplo, DisplayString se encuentra definido en SNMPv2-TC.

La solución alternativa es quitar o excluir la definición con problemas del archivo MIB.

[Tamaños alternativos](#)

Éste es sintácticamente un ejemplo válido, que indica que un valor del tipo `MyDatatype` será 0, 5, o 20 octetos de largo:

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

Algunos compiladores MIB no validan este sintaxis. Generalmente, un método alternativo aceptable es escoger uno de los tamaños y quitar los otros. Usted debe guardar el tamaño más grande. Por ejemplo, el ejemplo anterior sería cambiado a esto:

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

[Identificadores de objetos impares](#)

Algunos OID se consideran impares porque no refieren a un nodo en el S I (como la mayoría de los `identificadores de objeto`). Sin embargo, son sintácticamente válidos. Un ejemplo común es el `identificador de objeto nulo`, por ejemplo, `{0 0}`. Algunos compiladores MIB no cuidan para los `identificadores de objeto` que no corresponden a un nodo en el S I. Éstos son los ejemplos de sintaxis MIB que podrían causar los problemas para estos compiladores:

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

La solución alternativa es quitar o comentar hacia fuera esos tipos de referencias en el archivo MIB.

[Definiciones de notificaciones de trampa](#)

En el MIB del SNMPv1, los desvíos se definen con la macro `TRAP-TYPE`. En el MIB SNMPv2, los desvíos se definen usando la macro `NOTIFICATION-TYPE`.

Algunos compiladores MIB no les gusta estas definiciones en los archivos MIB que están analizando (no soportan estas macros).

Si éste es el caso, usted puede quitar las definiciones de trampa o comentar hacia fuera las definiciones (por ejemplo, ponga el delimitador de comentarios MIB -- al principio de las líneas).

[Compiladores basado en RFC 14XX contra los compiladores basado en RFC 19XX](#)

El RFCs 1442 define hasta el 1452 el SNMPv2 partido-basado. Estos RFC obsoleted por el RFCs 1902 más nuevo del estándar de borrador con 1908.

En lo que respecta a la sintaxis MIB, hay muy pocas diferencias entre estas dos versiones de SNMPv2; hay algo, sin embargo. El MIB de Cisco se basa actualmente en las reglas del RFC 19xx.

Note: Hace unos años, cuando el MIB de Cisco era RFC 14xx-based, algunos compiladores basado en RFC 19XX se quejarían por el `Unsigned32:: = línea TEXTUAL-CONVENTION` en el MIB CISCO-TC.my y PNNI-MIB.my. Esto es porque el Unsigned32 es un datatype predefinido en el RFC 19xx. Por este motivo, Cisco tenía versiones alternativas de este MIB (CISCO-TC-NO-U32.my y PNNI-MIB-NO-U32.my) sin la definición para el Unsigned32, a cargar en los compiladores que saben ya sobre este tipo de datos. Esto es no más aplicable.

[MIB que carga y de compilación en el NMS de tercera persona](#)

El mejor y la mayoría de la forma eficiente de cargar el MIB, los desvíos, y los iconos de Cisco en el NMS de tercera persona es utilizar la utilidad de integración de los CiscoWorks (utilidad de integración), que es disponible como parte del CiscoWorks Common Services (o independiente de <http://www.cisco.com/cgi-bin/tablebuild.pl/cw2000-utility>), con el adaptador correspondiente de la utilidad de integración de <http://www.cisco.com/tacpage/sw-center/cw2000/cmc3rd.shtml> y del último paquete de datos de la integración de administración de red (NMIDB). Documentación de la utilidad de integración del control para más detalles.

Alternativamente, usted puede consultar la documentación del NMS de tercera persona en el cargamento y la compilación MIB. Este documento incluye las instrucciones para el HP OpenView y el IBM NetView; pero usted debe todavía consultar HP o la documentación de IBM, pues los Productos pueden cambiar.

[Del GUI del HP OpenView o del IBM NetView](#)

Siga los siguientes pasos para cargar el MIB de Cisco que usted quiere:

1. Copie los archivos en el directorio `/usr/OV/snmp_mibs` de la estación de administración de red. Éste es el directorio predeterminado donde el HP OpenView y el IBM NetView buscan los documentos MIB. Si los coloca en cualquier otro lugar, especifique los nombres de trayecto explícitos en la interfaz gráfica loadmib.
2. Fije los permisos de modo que usted tenga acceso de lectura al MIB.
3. Del menú GUI, elija las **opciones > MIB de carga y descarga**.
4. Siga las instrucciones en la documentación de la plataforma, de compilar o de cargar el MIB de Cisco.

[De la interfaz de línea de comando del HP OpenView o del IBM NetView](#)

Publique `/opt/OV/bin/xnmloadmib` - cargue el *comando filename*, de cargar el archivo MIB.

Información Relacionada

- [Notas técnicas SNMP](#)
- [Soporte Técnico y Documentación - Cisco Systems](#)