MIB-Compiler und Laden von MIBs

Inhalt

Einführung

Die meisten Netzwerkmanagementsysteme (NMSs) bieten eine Möglichkeit zum Laden von MIBs. Beim Laden einer MIB kann ein NMS Informationen über neue MIB-Objekte wie deren Namen, Objekt-IDs (OIDs) und die Art des Datentyps (z. B. `Counter`) erhalten.

Die MIB kann beim Laden geparst werden oder später, z. B. beim Ausführen einer NMS-Anwendung. Die Software, die die Analyse durchführt, ist ein MIB-Compiler.

Alle syntaktisch korrekten MIBs sollten vom MIB-Compiler eines beliebigen Anbieters erfolgreich analysiert werden. Leider können verschiedene MIB-Compiler unterschiedliche Quirks aufweisen.

Cisco bemüht sich fortlaufend, sicherzustellen, dass die an Kunden veröffentlichten MIBs syntaktisch korrekt sind. Cisco vermeidet zudem MIB-Konstrukte, die sich bei gängigen NMS-Produkten als problematisch erwiesen haben. Trotz dieser Bemühungen ist es nicht möglich, die Quirks in allen MIB-Compilern auf dem Feld zu befriedigen.

In diesem Dokument werden einige häufige Probleme behandelt und Problemumgehungen vorgeschlagen. Wenn bei dem MIB-Compiler Ihres Anbieters eines dieser Probleme auftritt (mit Ausnahme des Problems RFC 14xx im Vergleich zu RFC 19xx), liegt dies an einem Mangel in diesem MIB-Compiler. Möglicherweise möchten Sie den Anbieter oder die Lieferanten auffordern,

ihre Compiler zu reparieren.

Voraussetzungen

Anforderungen

Die Leser dieses Dokuments sollten mit MIBs vertraut sein.

Verwendete Komponenten

Dieses Dokument ist nicht auf bestimmte Software- und Hardware-Versionen beschränkt.

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netz Live ist, überprüfen Sie, ob Sie die mögliche Auswirkung jedes möglichen Befehls verstehen.

Konventionen

Weitere Informationen zu Dokumentkonventionen finden Sie unter Cisco Technical Tips Conventions (Technische Tipps von Cisco zu Konventionen).

Häufige MIB-Ladeprobleme

Bestellung laden

Die Ladereihenfolge ist das wichtigste und häufigste Problem beim Laden von MIBs. Viele MIBs verwenden Definitionen, die in anderen MIBs definiert sind. Diese Definitionen sind in der `IMPORTS`-Klausel am oberen Rand der MIB aufgeführt.

Wenn beispielsweise MIB-**Mulbel** eine Definition aus MIB-**Stoßstangen** importiert, erfordern einige MIB-Compiler, dass Sie MIB-**Stoßstangen** vor dem Laden von MIB-**Mulbel** laden. Wenn Sie die Ladereihenfolge falsch verstehen, behauptet der Compiler, dass die importierten MIBs nicht definiert sind.

Dies ist eine Liste der MIBs, aus denen viele andere MIBs importieren, sowie die Reihenfolge, in der sie geladen werden sollten. Dies übernimmt wahrscheinlich 95 Prozent der Probleme mit der Lastverteilung (die meisten anderen MIBs können in beliebiger Reihenfolge geladen werden):

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

Hinweis: Wenn Sie die v1-Versionen dieser MIBs laden, sieht der MIB-Dateiname tatsächlich wie

IF-MIB-**V1SMI**.my ("-V1SMI") aus, der dem Namen der MIBs hinzugefügt wird, die von v2 in v1 konvertiert wurden. Eine Ausnahme bildet die RFC1213-MIB.my MIB, die nur als v1-Version existiert (d. h. es gibt kein RFC1213-MIB-V1SMI.my).

Wenn Sie versuchen, eine weitere MIB zu laden, und wenn der Compiler über nicht definierte Elemente beschwert, geben Sie an, von welchen MIBs diese MIB importiert wird, und überprüfen Sie, ob Sie zuerst alle anderen MIBs geladen haben.

Hinweis: Für jede MIB können Sie unter SNMP Object Navigator > View & Download MIBs die genaue Liste der MIBs anzeigen, die vor der MIBs geladen werden müssen, mit der genauen Kompilierungsreihenfolge. Wählen Sie **MIB-Abhängigkeiten anzeigen aus, und laden Sie MIB herunter**.

Diskrepanzen bei Datatype-Definitionen

Obwohl die Cisco MIB-Datentypdefinitionen nicht in etwa übereinstimmen, werden Sie dies bei einigen Standard-RFC-MIBs möglicherweise feststellen. Beispiele:

- MIB-**Multicast** definiert: `SomeDatatype ::= INTEGER(0.100)`
- MIB-**Humble** definiert: `SomeDatatype ::= INTEGER(1..50)`

Dieses Beispiel gilt als trivialer Fehler, und die MIB wird erfolgreich mit einer Warnmeldung geladen.

Das nächste Beispiel gilt als nicht trivialer Fehler (obwohl die beiden Definitionen im Wesentlichen äquivalent sind), und die MIB wird nicht erfolgreich analysiert.

- MIB-**Multicast** definiert: `SomeDatatype ::= DisplayString`
- MIB-**Humble** definiert: `SomeDatatype ::= OCTET STRING (SIZE(0.255))`

Wenn der MIB-Compiler diese als Fehler behandelt oder die Warnmeldungen entfernt werden sollen, bearbeiten Sie eine der MIBs, die denselben Datentyp definieren, sodass die Definitionen übereinstimmen.

Objekt-ID-Definitionen

Wenn Sie diese MIBs laden, können Sie OID-Neudefinitionen feststellen (es kann jedoch auch andere Fälle geben, in denen dieser Fehler auftritt):

- ALD-CISCO-CPU-MIB.my
- ALD-CISCO-ENV-MIB.my
- ALD-CISCO-MEMORY-MIB.my
- ALD-CISCO-SYSTEM-MIB.my

Beispiele:

- ALD-CISCO-CPU-MIB.my definiert: `lcpu OBJECT IDENTIFIER ::= { local 1 }`
- ALD-CISCO-ENV-MIB.my definiert: `lenv OBJECT IDENTIFIER ::= { local 1 }`

Wenn Sie diese beiden MIBs laden, kann der MIB-Compiler Beschwerden über die Neudefinierung des `lcpu-OBJECT-IDENTIFIERS` mit einem neuen Namen `lenv` einreichen. Die OLD-CISCO-MEMORY-MIB.my und OLD-CISCO-SYSTEM-MIB.my geben in ähnlicher Weise neue Namen für `{ local 1 }`.

Dies wird als trivialer Fehler behandelt, und die MIB wird erfolgreich mit einer Warnmeldung geladen.

Wenn die MIB nicht erfolgreich geladen wird oder die Warnmeldung entfernt werden soll, bearbeiten Sie eine der MIBs, sodass alle MIBs denselben Namen verwenden.

Definitionen integrierter Datentypen

Viele MIB-Compiler verfügen über integrierte Kenntnisse einiger Datentypen, z. B. DisplayString. Einige dieser Compiler beschweren sich, wenn sie eine Definition für diese Datentypen in einer MIB sehen. DisplayString ist beispielsweise in SNMPv2-TC definiert.

Die Problemumgehung besteht darin, die fehlerhafte Definition in der MIB-Datei zu entfernen oder herauszukommentieren.

Alternative Größen

Dies ist ein syntaktisch gültiges Beispiel, das angibt, dass ein Wert des Typs `MyDatatype` entweder 0, 5 oder 20 Oktette lang ist:

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

Einige MIB-Compiler akzeptieren diese Syntax nicht. In der Regel besteht eine ausreichende Problemumgehung darin, eine der Größen auszuwählen und die anderen zu entfernen. Sie sollten die größte Größe behalten. Im vorherigen Beispiel wird beispielsweise Folgendes geändert:

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

Odd-Objektbezeichner

Einige OIDs gelten als ungerade, da sie nicht auf einen Knoten im SMI verweisen (wie die meisten `OBJECT` IDENTIFIERs). Sie sind jedoch syntaktisch gültig. Ein häufiges Beispiel ist der NULL-Objekt-Bezeichner, z. B. `{ 0 }`. Einige MIB-Compiler kümmern sich nicht um `OBJECT` IDENTIFIERs, die keinem Knoten in SMI entsprechen. Dies sind Beispiele für MIB-Syntax, die für diese Compiler Probleme verursachen könnten:

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

Die Problemumgehung besteht darin, diese Verweistypen in der MIB-Datei zu entfernen oder herauszukommentieren.

Trap-Definitionen

In SNMPv1-MIBs werden Traps mit dem `TRAP-TYPE`-Makro definiert. In SNMPv2-MIBs werden Traps mithilfe des `NOTIFICATION-TYPE`-Makros definiert.

Einige MIB-Compiler mögen diese Definitionen nicht in den MIB-Dateien, die sie analysieren (sie unterstützen diese Makros nicht).

In diesem Fall können Sie die Trap-Definitionen entfernen oder die Definitionen kommentieren (z. B. das MIB-Kommentartrennzeichen - am Anfang der Zeilen).

Auf RFC 14xx basierende Compiler im Vergleich zu auf RFC 19xx basierenden Compilern

Die RFCs 1442 bis 1452 definieren das parteibasierte SNMPv2. Diese RFCs werden durch die neueren Draft Standard RFCs 1902 bis 1908 ersetzt.

Hinsichtlich der MIB-Syntax gibt es nur sehr wenige Unterschiede zwischen diesen beiden Versionen von SNMPv2. Es gibt jedoch einige. Cisco MIBs basieren derzeit auf RFC 19xx-Regeln.

Hinweis: Vor einigen Jahren, als Cisco MIBs auf RFC 14xx basieren, klagten einige RFC 19xx-basierte Compiler über die `Unsigned32 ::= TEXTUAL-CONVENTION`-Linie in CISCO-TC.my und PNI-MIB.my MIBs. Dies liegt daran, dass Unsigned32 ein vordefinierter Datentyp in RFC 19xx ist. Aus diesem Grund hatte Cisco bisher alternative Versionen dieser MIBs (CISCO-TC-NO-U32.my und PNI-MIB-NO-U32.my), ohne Definition für Unsigned32, zum Laden in Compiler, die bereits über diesen Datentyp Bescheid wissen. Dies gilt nicht mehr.

Laden und Kompilieren von MIBS in NMS von Drittanbietern

Die beste und effizienteste Methode zum Laden von Cisco MIBs, Traps und Symbolen in ein NMS eines Drittanbieters ist das CiscoWorks Integration Utility (Integration Utility), das als Teil von CiscoWorks Common Services (oder als Standalone-Version von http://www.cisco.com/cgi-bin/tablebuild.pl/cw2000-utility) mit dem entsprechenden Integration Utility Adapter von http://www.cisco.com/tacpage/sw-center/cw2000/cmc3rd.shtml verfügbar ist und dem neuesten Network Management Integration Data Bundle (NMIDB). Weitere Informationen finden Sie in der Dokumentation zum Integration Utility.

Alternativ können Sie die Dokumentation von Drittanbieter-NMS zum Laden und Kompilieren von MIBs konsultieren. Dieses Dokument enthält Anweisungen für HP OpenView und IBM NetView. Sie sollten sich jedoch weiterhin die Dokumentation von HP oder IBM ansehen, da sich die Produkte möglicherweise ändern.

Über die Benutzeroberfläche von HP OpenView oder IBM NetView

Gehen Sie folgendermaßen vor, um die gewünschten Cisco MIBs zu laden:

1. Kopieren Sie die Dateien in das Verzeichnis /usr/OV/snmp_mibs der Netzwerkmanagement-Station.Dies ist das Standardverzeichnis, in dem HP OpenView und IBM NetView nach MIB-Dokumenten suchen. Wenn Sie sie an einem anderen Ort platzieren, geben Sie die expliziten Pfadnamen in der grafischen Schnittstelle loadmib an.
2. Legen Sie die Berechtigungen so fest, dass Sie Lesezugriff auf die MIBs haben.
3. Wählen Sie im GUI-Menü **Optionen > MIBs laden/entladen aus**.
4. Befolgen Sie die Anweisungen in der Plattformdokumentation, um die Cisco MIBs zu kompilieren oder zu laden.

Über die Befehlszeilenschnittstelle von HP OpenView oder IBM NetView

Geben Sie den Befehl **/opt/OV/bin/xnmloadmib-load *filename* ein, um die MIB-Datei zu laden.**

Zugehörige Informationen

- Technischer Support und Dokumentation für Cisco Systeme