

# MIB コンパイラと MIB のロード

## 内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[表記法](#)

[MIB ロードに関する一般的な問題](#)

[ロード順序](#)

[データタイプ定義の不一致](#)

[オブジェクト識別子の再定義](#)

[組み込みデータタイプの定義](#)

[代替サイズ](#)

[特殊なオブジェクト識別子](#)

[トラップの定義](#)

[RFC 14xx ベースのコンパイラと RFC 19xx ベースのコンパイラ](#)

[サードパーティ NMS への MIB のロードとコンパイル](#)

[HP OpenView または IBM NetView の GUI を使用する場合](#)

[HP OpenView または IBM NetView のコマンドライン インターフェイスを使用する場合](#)

[関連情報](#)

## 概要

大部分のネットワーク管理システム ( NMS ) では、MIB をロードする手段がユーザに提供されています。MIB をロードすると、NMS では名前、オブジェクト識別子 ( OID )、データタイプの種類 ( 例 : Counter ) といった新しい MIB オブジェクトの詳細の確認ができます。

MIB の解釈は、ロード時か、後で NMS アプリケーションを実行するときなどに行われます。解析を実行するソフトウェアは、MIBコンパイラです。

構文的に正しいすべての MIB は、どのベンダーの MIB コンパイラでも正常に解釈できます。ただし、MIB コンパイラによって固有の傾向が異なる場合があります。

Cisco では、継続的に、正しい構文の MIB をお客様に公開するように努めています。また、一般的な NMS 製品で問題が明らかになっている MIB 構造は避けています。このような努力を払ってはおりますが、市販の MIB コンパイラ固有の傾向に完全に対応することはできません。

このドキュメントでは、一般的な問題の一部に対応し、回避策を提案しております。ご使用ベンダーの MIB コンパイラでこれらの問題 ( [RFC 14xx と RFC 19xx の問題を除く](#) ) が発生する場合、[その MIB コンパイラの不具合が原因です](#)。ベンダーに対してコンパイラの修正を依頼するように推奨いたします。

# [前提条件](#)

## [要件](#)

このドキュメントの読者は、MIB について理解している必要があります。

## [使用するコンポーネント](#)

このドキュメントの内容は、特定のソフトウェアやハードウェアのバージョンに限定されるものではありません。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。対象のネットワークが稼働中である場合には、どのようなコマンドについても、その潜在的な影響について確実に理解しておく必要があります。

## [表記法](#)

ドキュメント表記の詳細は、「[シスコテクニカルティップスの表記法](#)」を参照してください。

# [MIB ロードに関する一般的な問題](#)

## [ロード順序](#)

ロード順序は、MIB のロードにあたって最も重要でよく発生する問題です。多くの MIB では、別の MIB に定義されている定義が使用されます。これらの定義は、MIB の先頭近くにある `IMPORTS`

たとえば、MIB `mumble` が MIB `bumble` から定義をインポートする場合、一部の MIB コンパイラでは MIB `bumble` を MIB `mumble` よりも先にロードする必要があります。ロード順序を間違えると、インポートされた MIB が未定義であるとコンパイラから警告を受けます。

以下に示すのは、他の多くの MIB からインポートされる MIB のリストであり、この順序に従ってロードする必要があります。このリストで、ロード順序の問題の 95 % は解決できるはずですが（その他の大部分の MIB は任意の順序でロードできます）。

- 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

注：これらの MIB の v1 バージョンをロードする場合、MIB ファイル名は実際には IF-MIB-V1SMI.my のように表示されます（「-V1SMI」は v2 から v1 に変換された MIB の名前に追加追加されます）。この例外は、[RFC1213-MIB.my](#) という MIB です。この MIB は v1 バージョンのみで存在します（つまり、[RFC1213-MIB-V1SMI.my](#) は存在しません）。

他の MIB をロードしようとして、コンパイラから未定義のアイテムについて警告される場合には、この MIB でのインポート元の MIB を特定し、それ以外のすべての MIB が先にロードされていることを確認してください。

注：各MIBについては、[\[SNMP Object Navigator\] > \[View & Download MIBs\]](#)で、事前にロードする必要があるMIBの正確なリストを正確なコンパイル順序で確認できます。ここで、[\[View MIB dependencies and download MIB\]](#) を選択してください。

## データ タイプ定義の不一致

Cisco MIB データタイプ定義では不一致は発生しませんが、一部の標準 RFC MIB で発生することがあります。以下に、いくつかの例を示します。

- MIB mumble の定義 : `SomeDatatype ::= INTEGER(0..100)`
- MIB bumble の定義 : `SomeDatatype ::= INTEGER(1..50)`

この例はささいなエラーと見なされ、警告メッセージが表示されますが MIB は正常にロードされます。

次の例は、些細なエラーではありません (とはいえ、2つの定義は基本的に同じものです)。この場合、MIB は正常に解析されません。

- MIB mumble の定義 : `SomeDatatype ::= DisplayString`
- MIB bumble の定義 : `SomeDatatype ::= OCTET STRING (SIZE(0..255))`

ご使用の MIB コンパイラでこれらがエラーとして処理される場合や、警告メッセージが表示されないようにする場合は、この同じデータ タイプを定義している MIB のいずれかを編集し、定義を一致させてください。

## オブジェクト識別子の再定義

次の MIB をロードすると、OID 再定義が発生する場合があります (他の状況でもこのエラーが発生する場合があります)。

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

以下に、いくつかの例を示します。

- OLD-CISCO-CPU-MIB.my の定義 : `lcpu OBJECT IDENTIFIER ::= { local 1 }`
- OLD-CISCO-ENV-MIB.my の定義 : `lenv OBJECT IDENTIFIER ::= { local 1 }`

これら 2 つの MIB をロードすると、`lcpu OBJECT IDENTIFIER lenv` で再定義されている点について MIB コンパイラから警告される場合があります。OLD-CISCO-MEMORY-MIB.myとOLD-CISCO-SYSTEM-MIB.myも同様に、新しい名前を{ local 1}。

これはささいなエラーと見なされ、警告メッセージが表示されますが MIB は正常にロードされません。

MIB が正常にロードされない場合や、警告メッセージが表示されないようにする場合は、いずれかの MIB を編集し、すべての MIB で同じ名前を使用するようにしてください。

## 組み込みデータタイプの定義

多くの MIB コンパイラには、DisplayString といった一部のデータタイプがあらかじめ組み込まれています。これらのコンパイラの一部は、MIB にこれらのデータタイプの定義を発見すると警告を出します。たとえば、DisplayString は SNMPv2-TC で定義されています。

対策としては、MIB ファイル内で問題のある定義を削除するか、またはコメント化します。

## 代替サイズ

次の例はタイプ `MyDatatype 0 5 20`

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

一部の MIB コンパイラではこの構文は許容されません。一般的に有効な回避策としては、いずれかのサイズを選択し、他のサイズは削除します。最も大きいサイズを維持する必要があります。たとえば、前述の例を次のように変更します。

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

## 特殊なオブジェクト識別子

一部の OID は ( 大部分の OBJECT IDENTIFIER SMI ただし、これらは構文的には有効です。よくある例に、{ 0 0 } などの Null オブジェクト識別子があります。一部の MIB コンパイラでは、SMI 上のノードに対応しない OBJECT IDENTIFIER このようなコンパイラで問題が発生する可能性のある MIB 構文の例を次に示します。

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

回避策としては、MIB ファイルでこのようなタイプの参照を削除するか、コメント化します。

## トラップの定義

SNMPv1 MIB では、トラップは `TRAP-TYPE` SNMPv2 MIB では、`NOTIFICATION-TYPE`

一部の MIB コンパイラでは、解釈中の MIB ファイル中にこのような定義があると処理されません ( このようなコンパイラではこれらのマクロがサポートされていません )。

このような場合、トラップ定義を削除するか、定義をコメント化できます ( たとえば、MIB コメント デリミタ -- を行先頭に付加します )。

## RFC 14xx ベースのコンパイラと RFC 19xx ベースのコンパイラ

RFC 1442 ~ 1452 は、パーティベースの SNMPv2 を定義します。これらの RFC は、新しいドラフト標準 RFC 1902 ~ 1908 によって廃止されます。

これら 2 バージョンの SNMPv2 には、MIB 構文についてほとんど違いがありませんが、いくつかの差異は存在します。Cisco MIB は現在 RFC 19xx のルールに基づいています。

注：数年前に、Cisco MIBがRFC 14xxベースだった場合、一部のRFC 19xxベースのコンパイラでは、CISCO-TC.myおよびPNNI-MIB.my MIBのUnsigned32 ::= TEXTUAL-CONVENTION-MIBこれは、RFC 19xx の事前定義データタイプが Unsigned32 であるためです。この理由から、Cisco ではこれらの MIB について、Unsigned32 の定義が含まれない代替バージョン ( CISCO-TC-NO-U32.my および PNNI-MIB-NO-U32.my ) を、このデータタイプが認識されるコンパイラにロードするために準備していました。現在では、これは該当しません。

## サードパーティ NMS への MIB のロードとコンパイル

サードパーティ NMS への Cisco MIB、トラップ、およびアイコンのロードに最適で最も効率的な方法は、CiscoWorks Common Services の一部として入手できる ( または <http://www.cisco.com/cgi-bin/tablebuild.pl/cw2000-utility> から単体で入手できる ) CiscoWorks Integration Utility ( Integration Utility ) を、<http://www.cisco.com/tacpage/sw-center/cw2000/cmc3rd.shtml> から入手できる対応する Integration Utility Adapter、および最新のネットワーク管理統合データバンドル ( NMIDB ) とともに使用することです。詳細については、Integration Utility のドキュメントを確認してください。

あるいは、MIB のロードとコンパイルに関して、サードパーティ NMS のドキュメントを参照することもできます。このドキュメントでは、HP OpenView および IBM NetView での手順を説明しますが、これらの製品に変更が加えられている可能性があるため、HP や IBM のドキュメントも確認してください。

## HP OpenView または IBM NetView の GUI を使用する場合

次のステップに従い、必要な Cisco MIB をロードします。

1. ネットワーク管理ステーションの /usr/OV/snmp\_mibs ディレクトリにファイルをコピーします。これは、HP OpenView と IBM NetView が MIB ドキュメントを検索するデフォルトのディレクトリです。ドキュメントを別の場所に置く場合、loadmib グラフィカル インターフェイスで明示的なパス名を指定してください。
2. MIB に読み取りアクセスできるように権限を設定します。
3. GUI メニューから、[Options > Load/Unload MIBs] の順に選択します。
4. プラットフォームのドキュメントでの指示に従って、Cisco MIB のコンパイルやロードを行います。

## HP OpenView または IBM NetView のコマンドライン インターフェイスを使用する場合

/opt/OV/bin/xnmloadmib -load filename コマンドを発行し、MIB ファイルをロードします。

## 関連情報

- [テクニカル サポートとドキュメント – Cisco Systems](#)