CISCO-BULK-FILE-MIB の使用方法

内容

概要

このドキュメントでは、CISCO-BULK-FILE-MIB の使用方法と、この Management Information Base (MIB; 管理情報ベース) によって作成されたファイルを、CISCO-FTP-CLIENT-MIB を使用して転送する方法について説明します。

シスコでは、Cisco IOS® ソフトウェア リリース 12.0 から、Simple Network Management Protocol (SNMP; 簡易ネットワーク管理プロトコル) オブジェクトまたはテーブルをデバイス上のファイルとして保存する方法を実装しています。このファイルは、CISCO-FTP-CLIENT-MIB を使用して取得できます。このテクノロジーにより、信頼性の高い転送方式を使用して大量のデータを転送することが可能です。

はじめに

要件

この設定を開始する前に、次の要件が満たされていることを確認してください。

- 使用している Cisco デバイスで Cisco IOS® ソフトウェア リリース 12.0 以降が動作していること。また、MIB Locator ツールを使用して、そのデバイスで CISCO-BULK-FILE-MIB がサポートされていることを確認します。このツールへのリンクについては、「Cisco IOS MIB ツール」ページを参照してください。**注：このMIBは、Catalyst OSデバイスではサポートされていません。**

- デバイスに SNMP が読み取り専用および読み書き用コミュニティ ストリングとともに設定されていること。これについての説明は、このドキュメントの対象外です。IOS(R) デバイスでの SNMP の設定方法については、「ルータ、Cisco IOS ソフトウェア ベースの XL スイッチ、RSM、MSFC、および Catalyst スイッチで SNMP コミュニティ ストリングを設定する方法」を参照してください。

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- 12.1 (12) が動作している 7507 ルータの ifTable をファイルに保存するために CISCO-BULK-FILE-MIB を使用し、次にそのファイルをルータから FTP サーバに転送するために CISCO-FTP-CLIENT-MIB を使用します。
- UNIXまたはWindowsにインストールさ 、net-snmp SNMPコマンドスイート。
- 次の MIB を使用します。SNMPv2-TCSNMPv2-SMISNMPv2-CONFSNMPv2-MIBIANAifType-MIBIF-MIBCISCO-SMICISCO-TCCISCO-BULK-FILE-MIBCISCO-FTP-CLIENT-MIB

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実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。

背景説明

管理プラットフォームに次のテーブルの MIB がロードされていることを確認します。これにより、数値の Object Identifier (OID; オブジェクト識別子) の代わりに、上記のオブジェクト名と値を使用できます。 このドキュメントでは全般的に、OID ではなくオブジェクト名を使用しています。

バージョン 1 の SMI 形式	バージョン 2 の SMI 形式
SNMPv2-SMI-V1SMI.my	SNMPv2-SMI.my
SNMPv2-TC-V1SMI.my	SNMPv2-TC.my
	SNMPv2-CONF.my
SNMPv2-MIB-V1SMI.my	SNMPv2-MIB.my
IANAifType-MIB-V1SMI.my	IANAifType-MIB.my
IF-MIB-V1SMI.my	IF-MIB.my
CISCO-SMI-V1SMI.my	CISCO-SMI.my
CISCO-TC-V1SMI.my	CISCO-TC.my
CISCO-BULK-FILE-MIB-V1SMI.my	CISCO-BULK-FILE-MIB.my
CISCO-FTP-CLIENT-MIB-V1SMI.my	CISCO-FTP-CLIENT-MIB.my

CISCO-BULK-FILE-MIB の使用

BULK-FILE 処理の作成

この例では、ルータから ifTable を取得し、それをバルク ファイルに保存します。ただし、任意の MIB オブジェクトまたはテーブルを使用できます。

snmpset の net-snmp バージョンを使用します。ルータのIPアドレスは**14.32.8.2**で、読み取り/書き込みコミュニティストリングは**privateです**。読み取り専用コミュニティ ストリングは public です。

新規のバルク ファイル処理を作成するたびに、行インスタンス用の 2 つの乱数を選択します。使用可能な範囲は 1 ~ 4294967295 です。この例では、333 と 444 を使用します。

手順ごとの説明

BULK-FILE オペレーションを作成するには、次の手順を実行します。

1. 作成するファイルを設定します。
   ```
   $ snmpset -c private 14.32.8.2 cbfDefineFileEntryStatus.333 i 5
   $ snmpset -c private 14.32.8.2 cbfDefineFileName.333 s ifTable.txt
   $ snmpset -c private 14.32.8.2 cbfDefineFileFormat.333 i bulkASCII
   ```

2. 取得する MIB オブジェクトを指定します。正しく処理するためには、このオブジェクトに 2 つのインデックスが必要です。333 は、上記のファイル作成テーブルで使用した 333 です。444 は、cbfDefineObjectTable のプライマリ インデックスに使用する新規の乱数です。次のコマンドでは、cbfDefineObjectID (ifTable) のオブジェクト名を使用しています。 ここでは、完全な OID を使用することもできます。
   ```
   $ snmpset -c private 14.32.8.2 cbfDefineObjectID.333.444 o ifTable
   ```

3. 新規に作成した行をアクティブにします。cbfDefineObjectTable の行には両方のインデックスが必要です。
   ```
   $ snmpset -c private 14.32.8.2 cbfDefineObjectEntryStatus.333.444 i 1
   $ snmpset -c private 14.32.8.2 cbfDefineFileEntryStatus.333 i 1
   ```

4. ファイルを作成します。
   ```
   $ snmpset -c private 14.32.8.2 cbfDefineFileNow.333 i 3
   ```

   バルク ファイルが作成されます。
5. cbfStatusFileState オブジェクトに対して snmpget を使用して、ファイルが正常に作成されたことを確認します。このオブジェクトには 2 つのインデックスが必要です。最初のインデックスは、File テーブル用に選択した乱数 (この例では 333) です。 2 番目のインデックスは、ルータで作成したファイルの数によって異なります。これは最初のファイルなので、インデックスは1です。したがって、次のコマンドを使用します。
   ```
   $ snmpget -c public 14.32.8.2 cbfStatusFileState.333.1
   ```

   値 running(1) は、ファイルが現在作成中であることを意味します。値 ready(2) は、ファイルの作成が正常に終了し、ファイルが読み取り可能な状態にあることを意味します。

ただし、このファイルにはルータから直接アクセスできません。このファイルを読み取るには、CISCO-FTP-CLIENT-MIB を使用します。

CISCO-FTP-CLIENT-MIB を使用したファイル転送

FTP クライアント処理ごとに、行インスタンス用の乱数を選択する必要があります。上記の処理で使用したものと同じ乱数を使用できます。この例では、555 を使用します。

手順ごとの説明

CISCO-FTP-CLIENT-MIB を使用してファイルを転送するには、次の手順を実行します。

1. FTP クライアントの行インスタンスを作成します。
   ```
   $ snmpset -c private 14.32.8.2 cfcRequestEntryStatus.555 i 5
   ```

2. 必要なパラメータを設定します。LocalFile は上記の手順で作成したファイルと同じ名前にする必要があります。bulkASCII ファイルを転送するために putASCII を使用します。上記の手順で cbfDefineFileFormat を bulkBinary に設定した場合は、cfcRequestOperation を putBinary に設定する必要があります。
   ```
   $ snmpset -c private 14.32.8.2 cfcRequestOperation.555 i putASCII
   $ snmpset -c private 14.32.8.2 cfcRequestLocalFile.555 s ifTable.txt
   $ snmpset -c private 14.32.8.2 cfcRequestRemoteFile.555 s /home/Marcus/ifTable.txt
   $ snmpset -c private 14.32.8.2 cfcRequestServer.555 s 172.18.123.33
   $ snmpset -c private 14.32.8.2 cfcRequestUser.555 s Marcus
   $ snmpset -c private 14.32.8.2 cfcRequestPassword.555 s marcus123
   ```

3. 行をアクティブに設定して転送を開始します。
   ```
   $ snmpset -c private 14.32.8.2 cfcRequestEntryStatus.555 i 1
   ```

   FTP 転送が始まります。完了すると、ファイルは /home/Marcus/ifTable.txt に保存されます。
4. FTP 転送のステータスを取得するには、cfcRequestResult オブジェクトに対して再度 snmpget を使用します。このオブジェクトは、他の FTP オブジェクトで使用したものと同じインデックスを使用します。
   ```
   $ snmpget -c public 14.32.8.2 cfcRequestResult.555
   ```

   値 pending(1) は、ファイルがまだ転送中であることを意味します。値 success(2) は、ファイル転送が正常に終了したことを意味します。他の値はすべてエラーを示します。
5. ファイル転送が完了したら、再度 cbfStatusFileState オブジェクトの snmpget を実行します。今度は異なる値が得られます。
   ```
   $ snmpget -c public 14.32.8.2 cbfStatusFileState.333.1
   enterprises.cisco.ciscoMgmt.ciscoBulkFileMIB.ciscoBulkFileMIBObjects.cbfStatus.
   cbfStatusFileTable.cbfStatusFileEntry.cbfStatusFileState.333.1 = emptied(3)
   ```
   値 emptied(3) は、ファイルの読み取りが正常に終了したことを意味します。このファイルは再転送できません。
6. この状態になれば、ファイル ステータス行を破棄して、このファイルを削除しても安全です。このオブジェクトは、上記の cbfStatusFileState と同じインデックスをとります。
   ```
   $ snmpset -c private 14.32.8.2 cbfStatusFileEntryStatus.333.1 i 6
   ```

7. ファイルが削除されたら、対応する Object と File の行も削除します。
   ```
   $ snmpset -c private 14.32.8.2 cbfDefineObjectEntryStatus.333.444 i 6
   $ snmpset -c private 14.32.8.2 cbfDefineFileEntryStatus.333 i 6
   ```

このように CISCO-FTP-CLIENT-MIB を使用して、FTP 経由でルータから任意のファイルを転送できます。

結果の確認

この項では、このファイルの構文の一部を解釈するためのガイドを示します。

1. 最初の行は prefix 行です。ifTable の例では、これは次のようになっています。
   ```
   prefix 1.3.6.1.2.1.2.2.1
   ```
   これは ifEntry オブジェクトの OID に対応します。ifTable は 1 つ以上の ifEntry から構成されます。
2. 次の行は、テーブルに含まれるオブジェクトの数を列挙しています。この行は table キーワードから始まり、その後にテーブルに含まれるオブジェクトの数と各オブジェクトのインデックスが続きます。以下に、いくつかの例を示します。
   ```
   table 22 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
   ```
   この行は、テーブル内に 22 個のオブジェクトがあり、各オブジェクトのインデックスが 1 刻みで増加していることを示しています。`ifTable`
   ```
   ifIndex
   ifDescr
   ifType
   ifSpeed
   ...
   ```
3. この行の後に複数の行エントリが続きます。ifTable の例では、各行は 1 つのインターフェイスに対応します。行は row キーワードから始まり、その後に行のインデックス ID と、前のテーブル エントリに列挙されたオブジェクトが続きます。以下に、いくつかの例を示します。
   ```
   row 1 1 546F6B656E52696E67302F30 9 4464 16000000 0008B0851800 2 2 6551 0 0 0 0 0 0 0 0 0 0 0 0.0
   ```
4. 4番目のエントリはイース1のifDescrです。ただし、これは16進数でエンコードされたASCIIの`ifDescr`です。この行を判読可能な形式に変換するには、次の Perl コマンドを使用します。
   ```
   $ perl -e 'print pack("H*", "546F6B656E52696E67302F30")'
   TokenRing0/0
   ```
   このエントリはインターフェイス TokenRing0/0 に対応します。バルク ファイルでは、通常、文字列であるオブジェクトはすべて、16 進数にエンコードされた ASCII で表示されます。この Perl コマンドは、どのような 16 進 ASCII 文字列でも判読可能なテキストに変換できます。Perlがない場合は、このASCII文字テーブルを使用して文字列を変換します。
5. エントリの中には、値として ~ 文字を示すものがあります。これは、そのオブジェクトの値が NULL であることを意味します。つまり、そのオブジェクトはデバイス上でインスタンス化されていません。以下に、いくつかの例を示します。
   ```
   row 9 9 41544D312F302F302D61746D206C61796572 37 ~ 0  1 1 5971 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
   ```
   これは、ATM1/0/0 ATM 層インターフェイスに対応します。このインターフェイスでは、ifMtu が NULL になっています。これは仮想インターフェイスであるため、MTU がないことは理にかなっています。必要であれば、デバイスの設定に次のコマンドを追加することで、これらの NULL を **0 に置き換えることができます。**
   ```
   Router(config)#no snmp-server sparse-table
   ```

結果のトラブルシューティング

cbfStatusFileState オブジェクトをポーリングしたときに running(1)、ready(2)、または emptied(3) 以外の値を受け取った場合は、処理でエラーが発生しています。このエラーの原因は次のとおりです。

```
noSpace     no data due to insufficient file space
badName     no data due to a name or path problem
writeErr    no data due to fatal file write error
noMem       no data due to insufficient dynamic memory
buffErr     implementation buffer too small
aborted     short terminated by operator command
```

ファイル内のオブジェクトの数が予想よりも少ない場合は、CISCO-BULK-FILE-MIB の cbfDefineMaxObjects が非常に小さい値に設定されている可能性があります。このオブジェクトの現在の値を確認するには、snmpget を使用します。

```
$ snmpget -c public 14.32.8.2 cbfDefineMaxObjects.0
```

値が 0 のときは制限が設定されていないことを意味します。この値は、0 ~ 4294967295 の範囲にある任意の整数に設定できます。ファイルあたりのオブジェクトの最大数を 10 に設定するには、**snmpset コマンドを使用します。**このオブジェクトのインデックスは常に 0 です。

```
$ snmpset -c private 14.32.8.2 cbfDefineMaxObjects.0 u 10
```

プラットフォームの中には、このオブジェクトを設定できないものがあります。**snmpset が次のエラーで失敗する場合は、そのプラットフォームで cbfDefineMaxObjects オブジェクトは設定できません。**

```
Error in packet.
Reason: (noSuchName) There is no such variable name in this MIB.
Failed object:
enterprises.cisco.ciscoMgmt.ciscoBulkFileMIB.ciscoBulkFileMIBObjects.cbfDefine.cbfDefineMaxObjects.0
```

cfcRequestResult オブジェクトをポーリングしたときに pending(1) または success(2) 以外の値を受け取った場合は、FTP 処理でエラーが発生しています。このエラーの原因は次のとおりです。

```
aborted              user aborted the transfer
fileOpenFailLocal    local bulk file was not found
fileOpenFailRemote   remote file could not be opened for writing
badDomainName        FTP server's hostname could not be resolved
unreachableIpAddress route to the FTP server could not be found
linkFailed           connection could not be made to the remote server
fileReadFailed       local file could not be read
fileWriteFailed      remote file could not be written
```

警告

- 現在、バルク ファイルに直接アクセスする方法はサポートされていません。このファイルを読み取るには、CISCO-FTP-CLIENT-MIB を使用する必要があります。
- `cbfDefineFileStorage ephemeral`、`volatile`、および `permanent` の 3 タイプを定義しています。現在 IOS でサポートされているタイプは ephemeral のみです。少量の短命ファイルが、読み取られるまで存在します。

- 一度読み取られたファイルを再度読み取ることはできません。最初から再作成する必要があります。
- `cbfDefineFileFormat standardBER`、`bulkBinary`、および `bulkASCII` の 3 タイプを定義しています。サポートされている形式は bulkBinary と bulkASCII のみです。デフォルトの形式は bulkBinaryです。
- Windows 用の Chameleon FTP サーバは正しい結果コードを返さないため、CISCO-FTP-CLIENT-MIB と組み合せて使用できないことがわかっています。

関連情報

- ルータ、Cisco IOS ソフトウェア ベースの XL スイッチ、RSM、MSFC、および Catalyst スイッチで SNMP コミュニティ ストリングを設定する方法
- テクニカルサポート - Cisco Systems