

如何使用CISCO-BULK-FILE-MIB

目錄

[簡介](#)

[開始之前](#)

[需求](#)

[採用元件](#)

[慣例](#)

[背景資訊](#)

[使用CISCO-BULK-FILE-MIB](#)

[建立BULK-FILE操作](#)

[逐步說明](#)

[使用CISCO-FTP-CLIENT-MIB傳輸檔案](#)

[逐步說明](#)

[驗證結果](#)

[對結果進行故障排除](#)

[注意事項](#)

[相關資訊](#)

簡介

本檔案將說明如何使用CISCO-BULK-FILE-MIB，以及如何使用CISCO-FTP-CLIENT-MIB傳輸該管理資訊庫(MIB)建立的檔案。

自Cisco IOS®軟體版本12.0開始，思科實作了一種將簡易網路管理通訊協定(SNMP)物件或表儲存為裝置上檔案的方法。然後可使用CISCO-FTP-CLIENT-MIB檢索此檔案。此技術允許您使用可靠的傳輸方法傳輸大量資料。

開始之前

需求

嘗試此設定之前，請確保符合以下要求：

- 您有一台運行Cisco IOS®軟體版本12.0或更高版本的Cisco裝置。檢查MIB Locator Tool，確保您的裝置支援CISCO-BULK-FILE-MIB。在[Cisco IOS MIB Tools](#)頁面可以找到該工具的連結。
註：Catalyst OS裝置不支援此MIB。
- 必須在裝置上使用只讀和讀寫社群字串配置SNMP。本檔案沒有說明這一點。有關在IOS®裝置上配置SNMP的資訊，請參閱[如何在路由器、基於Cisco IOS軟體的XL交換機、RSM、MSFC和Catalyst交換機上配置SNMP社群字串](#)。

採用元件

本文中的資訊係根據以下軟體和硬體版本：

- CISCO-BULK-FILE-MIB將來自運行12.1(12)的7507路由器的ifTable儲存在檔案中，然後使用CISCO-FTP-CLIENT-MIB將該檔案從路由器傳輸到FTP伺服器。
- 在UNIX或Windows上安裝[net-snmp](#) 的SNMP命令套件。
- 使用以下MIB:SNMPv2-TCSNMPv2-SMISNMPv2-CONF SNMPv2-MIBIANAifType-MIBIF-MIB CISCO-SMICISCO-TCCISCO-BULK-FILE-MIB CISCO-FTP-CLIENT-MIB

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路正在作用，請確保您已瞭解任何指令可能造成的影響。

慣例

如需文件慣例的詳細資訊，請參閱[思科技術提示慣例](#)。

背景資訊

請確保已將此表中的MIB載入到您的管理平台中。這允許您使用上面列出的對象名稱和值，而不是數字對象識別符號(OID)。通常，本文檔引用對象名稱而不是OID。

版本1 SMI格式	版本2 SMI格式
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

使用CISCO-BULK-FILE-MIB

建立BULK-FILE操作

在本例中，我們從路由器捕獲ifTable，並將其儲存在批次檔案中。但是，您可以使用任何MIB對象或表。

使用net-snmp版本的snmpset。路由器的IP地址為14.32.8.2。其讀寫社群字串為private。只讀社群字串為public。

每次建立新的批次檔案操作時，為行例項選擇兩個隨機數。它們可以是介於1和4294967295（包括

1和) 之間的任意數字。在本例中，使用333和444。

逐步說明

要建立BULK-FILE操作，請完成以下步驟：

1. 設定要建立的檔案。

```
$ 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. 指定要捕獲的MIB對象。此對象需要兩個索引才能正確操作。333是上述檔案建立表中的333。444是一個新的隨機數，用於cbfDefineObjectTable中的主。此示例演示使用cbfDefineObjectID(ifTable)。您也可以在這裡使用完全限定的OID。

```
$ snmpset -c private 14.32.8.2 cbfDefineObjectID.333.444 o ifTable
```

3. 啟用新建的行。必須為cbfDefineObjectTable行設定兩引。

```
$ snmpset -c private 14.32.8.2 cbfDefineObjectEntryStatus.333.444 i 1
$ snmpset -c private 14.32.8.2 cbfDefineFileEntryStatus.333 i 1
```

4. 建立檔案。

```
$ snmpset -c private 14.32.8.2 cbfDefineFileNow.333 i 3
```

建立批次檔案。

5. 使用cbfStatusFileState 對象上的snmpget驗證檔案是否建立成功。此對象需要兩個索引。第一個索引是為檔案表 (本例中為333) 選擇的隨機數。第二個索引取決於您在路由器中建立的檔案數。由於這是第一個檔案，因此索引為1。因此，請使用命令：

```
$ snmpget -c public 14.32.8.2 cbfStatusFileState.333.1
```

值running(1)表示檔案正在建立過程中。ready(2)檔案已成功建立，並且正在等待讀取。但是不能從路由器直接訪問此檔案。使用CISCO-FTP-CLIENT-MIB讀取此檔案。

使用CISCO-FTP-CLIENT-MIB傳輸檔案

對於每個FTP客戶端操作，必須為行例項選擇一個隨機數。您可以使用上面使用的同一隨機數之一。此示例使用555。

逐步說明

要使用CISCO-FTP-CLIENT-MIB傳輸檔案，請完成以下步驟：

1. 建立FTP客戶端的行例項。

```
$ snmpset -c private 14.32.8.2 cfcRequestEntryStatus.555 i 5
```

2. 填寫所需引數。LocalFile 必須與上面建立的檔案同名！使用putASCII傳輸bulkASCII 檔案。如果將cbfDefineFileFormat 設定為bulkBinary，則必須將cfcRequestOperation 設定為Binary。

```
$ 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. 通過將行設定為活動開始傳輸。

```
$ snmpset -c private 14.32.8.2 cfcRequestEntryStatus.555 i 1
```

FTP傳輸開始。完成後，檔案會儲存到/home/Marcus/ifTable.txt。

4. 要獲取FTP傳輸的狀態，請再次對cfcRequestResult對象使用snmpget。此對象使用與其他FTP對象相同的索引。

```
$ snmpget -c public 14.32.8.2 cfcRequestResult.555
```

值為pending(1)檔案仍在傳輸。值success(2)檔案傳輸成功。任何其它值都是錯誤。

5. 完成檔案傳輸後，請重試cbfStatusFileState對象的snmpget。現在它有了不同的價值。

```
$ snmpget -c public 14.32.8.2 cbfStatusFileState.333.1
enterprises.cisco.ciscoMgmt.ciscoBulkFileMIB.ciscoBulkFileMIBObjects.cbfStatus.
cbfStatusFileTable.cbfStatusFileEntry.cbfStatusFileState.333.1 = emptied(3)
```

emptied(3)的值表示檔案已成功讀取。無法再次傳輸檔案。

6. 現在，通過銷毀檔案狀態行可以安全地刪除此檔案。此對象採用與上面的cbfStatusFileState相同的索引。

```
$ snmpset -c private 14.32.8.2 cbfStatusFileEntryStatus.333.1 i 6
```

7. 刪除檔案後，刪除相應的「對象」和「檔案」行。

```
$ snmpset -c private 14.32.8.2 cbfDefineObjectEntryStatus.333.444 i 6
```

```
$ snmpset -c private 14.32.8.2 cbfDefineFileEntryStatus.333 i 6
```

通過這種方式，您可以使用CISCO-FTP-CLIENT-MIB使用FTP從路由器傳輸任何檔案。

驗證結果

本節將指導您閱讀此檔案的一些語法。

1. 第一行是行。對於ifTable示例，該示例為：

```
prefix 1.3.6.1.2.1.2.2.1
```

此值對應於ifEntry對象OID。IfTable由一個或多個ifEntries組成。

2. 下一行列出表中的對象數。該行由關鍵字表、表中的對象數以及每個對象的索引組成。例如：

```
table 22 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
```

此行宣告該表包含22個對象，每個對象都有一個遞增索引。這些對象來自ifTable例：

```
ifIndex
ifDescr
ifType
ifSpeed
...
```

3. 此行之後有多個行條目。在ifTable示例中，每行都對應一個介面。行以關鍵字行開頭，後跟其索引識別符號，後跟由上一個表條目列舉的對象。例如：

```
row 1 1 546F6B656E52696E67302F30 9 4464 16000000 0008B0851800 2 2 6551 0 0 0 0 0 0 0 0 0 0 0
0 0 0.0
```

4. 第四個條目是介面1的ifDescr。但是，這是十六進位制編碼ASCII中的ifDescr。若要將此行轉換為更易讀的格式，請使用以下Perl命令：

```
$ perl -e 'print pack("H*", "546F6B656E52696E67302F30")'
TokenRing0/0
```

此條目對應於介面TokenRing0/0。通常是字串的所有對象在批次檔案中顯示為十六進位制編碼的ASCII。您可以使用此Perl命令將任何十六進位制ASCII字串轉換為可讀文本。如果沒有Perl，請使用此[ASCII字元](#) 表轉換字串。

5. 某些條目顯示~個值字元。這意味著該對象的值為NULL。也就是說，對象不會在裝置上例項化

。例如：

```
row 9 9 41544D312F302F302D61746D206C61796572 37 ~ 0 1 1 5971 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
```

此對應於ATM1/0/0-atm層介面。請注意，此介面的IfMtu為NULL。由於這是一個虛擬介面，因此它沒有MTU很有意義。如果您願意，可以將此命令新增到裝置的配置中，以0替換這些NULL：

```
Router(config)#no snmp-server sparse-table
```

對結果進行故障排除

輪詢cbfStatusFileState對象時，如果您收到running(1)、ready(2)或emptied(3)以外的值，則操作遇到錯誤。以下是錯誤的原因：

```
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
```

如果檔案中的對象數量小於預期數量，則CISCO-BULK-FILE-MIB中的cbfDefineMaxObjects可能設定得太低。要確定對象的當前值，請使用snmpget。

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

值為0表示未配置任何限制。該值可以設定為0到4294967295 (包括) 之間的任何整數。要將每個檔案的最大對象數設定為10，請使用snmpset命令。此對象的索引始終為0。

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

此對象並非在所有平台上均可配置。如果snmpset失敗並出現此錯誤，則對象在您的平台上不可配置：

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

輪詢cfcRequestResult對象時，如果您收到pending(1)或success(2)以外的值，則FTP操作遇到錯誤。以下是錯誤的原因：

```
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
```

注意事項

- 目前沒有支援的方法直接存取批次檔案。您必須通過CISCO-FTP-CLIENT-MIB來讀取檔案。
- `cbfDefineFileStorage` 定義了三種型別：、易和永久。目前，IOS中支援的唯一型別是。在讀取之前，臨時檔案數量很小。
- 一旦檔案被讀取，便無法重新讀取。必須首先重新建立它們。
- `cbfDefineFileFormat` 定義三種型別：standardBER、bulkBinary和bulkASCII。唯一支援的格式為bulkBinary和bulkASCII。預設格式為bulkBinary。
- 眾所周知，Windows版Chameleon FTP伺服器不與CISCO-FTP-CLIENT-MIB一起使用，因為它沒有返回正確的結果代碼。

相關資訊

- [如何在路由器、基於Cisco IOS軟體的XL交換器、RSM、MSFC和Catalyst交換器上設定SNMP社群字串](#)
- [技術支援 - Cisco Systems](#)