

如何使用 CISCO-BULK-FILE-MIB

目录

[简介](#)

[开始使用前](#)

[要求](#)

[使用的组件](#)

[规则](#)

[背景信息](#)

[使用 CISCO-BULK-FILE-MIB](#)

[创建 BULK-FILE 操作](#)

[逐步指导](#)

[使用 CISCO-FTP-CLIENT-MIB 传送文件](#)

[逐步指导](#)

[验证结果](#)

[排除结果中显示的故障](#)

[注意事项](#)

[相关信息](#)

简介

本文档介绍如何使用CISCO-BULK-FILE-MIB，以及如何使用CISCO-FTP-CLIENT-MIB传输由该管理信息库(MIB)创建的文件。

从Cisco IOS®软件版本12.0开始，思科已实施一种方法，将简单网络管理协议(SNMP)对象或表存储为设备上的文件。然后，可使用CISCO-FTP-CLIENT-MIB检索此文件。此技术允许您使用可靠的传输方法传输大量数据。

开始使用前

要求

在尝试此配置前，请保证您符合这些要求：

- 您有一台运行Cisco IOS®软件版本12.0或更高版本的思科设备。检查MIB定位器工具，确保您的设备支持CISCO-BULK-FILE-MIB。在Cisco IOS MIB工具页上可以找[到该工具](#)的链接。**注意**：此MIB在Catalyst OS设备上不受支持。
- 必须在设备上配置SNMP，同时使用只读和读写团体字符串。本文档不涉及此内容。有关在IOS®设备上配置SNMP的信息，请阅读[How to Configure SNMP Community Strings on Routers, Cisco IOS Software-Based XL Switches, RSMs, MSFCs and Catalyst Switches](#)。

使用的组件

本文档中的信息基于以下软件和硬件版本：

- CISCO-BULK-FILE-MIB将运行12.1(12)的7507路由器的ifTable存储到文件中，然后使用CISCO-FTP-CLIENT-MIB将该文件从路由器传输到FTP服务器。
- 在UNIX或Windows上安装的 net-snmp SNMP命令套件。
- 使用以下MIB:SNMPv2-TCSNMPv2-SMISNMPv2-CONF SNMPv2-MIBIANAifType-MIBIF-MIB CISCO-SMI思科 — TCCISCO-BULK-FILE-MIB CISCO-FTP-CLIENT-MIB

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

规则

有关文件规则的更多信息请参见“Cisco技术提示规则”。

背景信息

确保此表中的MIB已加载到管理平台中。这允许您使用上面列出的对象名称和值，而不是数字对象标识符(OID)。一般而言，本文档是指对象名称，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

创建 BULK-FILE 操作

在本示例中，我们从路由器捕获ifTable，并将其存储在批量文件中。但是，可以使用任何MIB对象或表。

使用snmpset的net-snmp版本。路由器的IP地址是14.32.8.2。其读写社区字符串是私有。只读社区字符串为public。

每次创建新的批量文件操作时，为行实例选择两个随机数。它们可以是介于1和4294967295之间的

任意数字。在本例中，请使用333和444。

逐步指导

要创建BULK-FILE操作，请完成以下步骤：

1. 设置要创建的文件。

```
$ 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. 指定要捕获的MIB对象。此对象需要两个索引才能正确操作。333是上面文件创建表中的333。444是用于cbfDefineObjectTable中主索引的新值。此示例演示如何为cbfDefineObjectID(ifTable)。您也可以在此处使用完全限定的OID。

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

3. 激活新创建的行。必须为cbfDefineObjectTable行同时索引。

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

4. 创建文件。

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

批量文件即会创建。

5. 验证是否在cbfStatusFileState对象上使用snmpget成功创建。此对象需要两个索引。第一个索引是为文件表（本例中为333）选择的随机数。第二个索引取决于您在路由器中创建的文件数。因为这是您的第一个文件，所以索引是1。因此，请使用命令：

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

running(1)表示文件正在创建过程中。ready(2)表示文件已成功创建，正在等待读取。

但是，无法从路由器直接访问此文件。使用CISCO-FTP-CLIENT-MIB读取此文件。

使用 CISCO-FTP-CLIENT-MIB 传送文件

对于每个FTP客户端操作，必须为行实例选择一个随机数。您可以使用上面使用的随机数之一。本示例使用 555。

逐步指导

要使用CISCO-FTP-CLIENT-MIB传输文件，请完成以下步骤：

1. 创建FTP客户端的行实例。

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

2. 填写所需参数。LocalFile必须与您在上面创建的文件同名！使用putASCII传输批ASCII文件。如果将cbfDefineFileFormat设置为上述bulkBinary，则必须将cfcRequestOperation设置为

putBinary。

```
$ 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. 将行设置为活动，开始传输。

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

FTP传输开始。完成后，文件将保存到/home/Marcus/ifTable.txt。

4. 要获取FTP传输的状态，请在cfcRequestResult对象上再次使用snmpget。此对象使用与其他FTP对象相同的索引。

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

pending(1)示文件仍在传输。值(2)为传输的文件。任何其他值都是错误。

5. 当文件传输完成后，请再次尝试cbfStatusFileState对snmpget。它现在有了不同的价值。

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

空(3)示文件已成功读取。无法再次传输文件。

6. 现在，通过销毁文件状态行来删除此文件是安全的。此对象采用与上述cbfStatusFileState相引

o

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

7. 删除文件后，删除相应的“对象”行和“文件”行。

```
$ snmpset -c private 14.32.8.2 cbfDefineObjectEntryStatus.333.444 i 6
$ snmpset -c private 14.32.8.2 cbfDefineFileEntryStatus.333 i 6
```

通过这种方式，您可以使用CISCO-FTP-CLIENT-MIB使用FTP从路由器传输任何文件。

验证结果

本节将引导您阅读此文件的一些语法。

1. 第一行是前行。对于我ifTable示例，它是：

```
prefix 1.3.6.1.2.1.2.2.1
```

这对应于ifEntry对OID。ifTable由一个或多个ifEntries。

2. 下一行列出表中的对象数。该行由关键字表和表中的对象数组成，后跟每个对象的索引。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
```

此行表示表包含22个对象，每个对象都有递增索引。这些对象来自ifTable示例：

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

3. 在此行之后，有多个行条目。在ifTable示例中，每行都对应一个接口。这些行以关键字行开头，后跟索引标识符，后跟上一个表条目枚举的对象。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
```

4. 第四个条目是接口1ifDescrASCII中的ifDescr。要将此行转换为更易读的格式，请使用以下Perl命令：

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

此条目对应于接口TokenRing0/0。通常为字符串的所有对象在批量文件中显示为十六进制编码

的ASCII。您可以使用此Perl命令将任何十六进制ASCII字符串转换为可读文本。如果没有Perl，请使用此ASCII字符表 [\[1\]](#)来转换字符串。

5. 某些条目显示值的~字符。这意味着该对象的值为NULL。也就是说，对象未在设备上实例化。

例如：

```
row 9 9 41544d312f302f302d61746d206c61796572 37 ~ 0 1 1 5971 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
```

这对应于ATM1/0/0-atm层接口。请注意，ifMtu对于此接口为NULL。由于这是虚拟接口，因此它没有MTU是有意义的。如果您愿意，可以通过将以下命令添加到设备配置中，将这些NULL替换为0:

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

排除结果中显示的故障

当轮询cbfStatusFileState对象时，如果收到除运行(1)、(2)或(3)以外的值，则操作会遇到错误。以下是导致错误的原因：

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

如果文件中的对象数少于预期，则CISCO-BULK-FILE-MIB中的cbfDefineMaxObjects可能设置得太低。要确定对象的当前值，请使用snmpget。

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

值0表示未配置限制。值可设置为0到4294967295（包括0和）之间的任意整数。要将每个文件的最大对象数设置为10，请使用snmpset命令。此对象的索引始终为0。

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

此对象可能无法在所有平台上配置。如果snmpset因此错误而失败，则对象无法在您的平台上配置：

Error in packet.

Reason: (noSuchName) There is no such variable name in this MIB.

Failed object:

```
enterprises.cisco.ciscoMgmt.ciscoBulkFileMIB.ciscoBulkFileMIBObjects.cbfDefine.cbfDefineMaxObjects.0
```

轮询cfcRequestResult时，如果收到的值不是pending(1)success(2)FTP以下是导致错误的原因：

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

注意事项

- 目前没有直接访问批量文件的支持方法。您必须通过CISCO-FTP-CLIENT-MIB读取文件。
- cbfDefineFileStorage定义了三种类型：、易和永。目前，IOS中支持的唯一类型是。临时文件在读取前存在少量文件。
- 读取文件后，将无法重新读取。必须首先重新创建它们。
- cbfDefineFileFormat定义了三种类型：standardBER、bulkBinary和bulkASCII。唯一支持的格式bulkBinary和bulkASCII。默认格式为bulkBinary。
- Windows版Chameleon FTP服务器已知不能与CISCO-FTP-CLIENT-MIB配合使用，因为它不返回正确的结果代码。

相关信息

- [如何在路由器、基于软件的 Cisco IOS XL 交换机、RSM、MSFC 和 Catalyst 交换机中设定 SNMP 社区字符串](#)
- [技术支持 - Cisco Systems](#)