

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](#)

[警告](#)

[相关信息](#)

简介

本文解释如何使用CISCO-BULK-FILE-MIB和转接该管理信息库(MIB)创建的文件使用CISCO-FTP-CLIENT-MIB。

从Cisco IOS软件版本12.0开始，Cisco实现方式存储简单网络管理协议(SNMP)对象或表作为在设备的一个文件。使用CISCO-FTP-CLIENT-MIB，此文件可能然后被检索。使用可靠的传输方法，此技术允许您传递很多数据。

开始使用前

要求

在尝试此配置前，请保证您符合这些要求：

- 您有运行Cisco IOS软件版本12.0的一台Cisco设备或以后。检查MIB定位器工具确保CISCO-BULK-FILE-MIB为您的设备支持。对工具的一条链路可以在[Cisco IOS MIB工具](#)页找到。**注意**：Catalyst OS设备不支持此MIB。
- 在有只读和读写团体串的设备必须配置SNMP。这在本文没有被覆盖。关于配置在IOS®设备的SNMP的信息，读[如何配置在路由器的SNMP团体字符串，基于Cisco IOS软件的XL交换机、RSM、MSFC和Catalyst交换机](#)。

使用的组件

本文档中的信息基于以下软件和硬件版本：

- 存储从7507个路由器运行12.1(12)的ifTable的CISCO-BULK-FILE-MIB在文件，然后使用CISCO-FTP-CLIENT-MIB转接从路由器的该文件到FTP服务器。
- 在UNIX 或Windows安装的[net-snmp snmp命令](#)套件。
- 使用这些MIB：SNMPv2-TCSNMPv2-SMISNMPv2-CONFSNMPv2-MIBIANAifType-MIBIF-MIBCISCO-SMICISCO-TCCISCO-BULK-FILE-MIBCISCO-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

有关文档规则的详细信息，请参阅 [Cisco 技术提示规则](#)。

背景信息

确保您有MIB在此表里装载到您的管理平台。这允许您使用对象名和值以上所列而不是数字对象标识符(OIDs)。一般来说，本文是指而不是对象名OIDs。

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

在本例中，我们在一个大批文件捕获从路由器的ifTable，并且存储它。然而，您能使用所有MIB对象或表。

请使用snmpset net-snmp版本。路由器的IP地址是14.32.8.2。其读写团体串私有。只读属性字段公共。

每次您创建一新的大批文件操作，请选择行实例的两个随机数。他们可以是1和4294967295范围的

任何编号包括。为此示例，使用333和444。

逐步指导

要创建BULK-FILE操作，请完成这些步骤：

1. 设置将创建的文件。

```
$ 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MIB对象捕获。此对象要求正确操作的两个索引。333是333从上面文件创建表。444是用于主要索引的一个新的随机数在cbfDefineObjectTable。此示例展示使用一个对象名对于cbfDefineObjectID (ifTable)。您可能使用一完全合格的OID这里。

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

3. 激活新建的行。您必须有您的cbfDefineObjectTable行的两个索引。

```
$ 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
```

大批文件创建。

5. 通过使用在cbfStatusFileState snmpget验证文件顺利地创建。此对象要求两个索引。第一个索引是为文件表选择的随机数(333在本例中)。第二个索引取决于多少个文件您在您的路由器创建。因为这是您的第一个文件，索引是1。所以，请使用命令：

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

值running(1)意味着文件是在创建过程中。值ready(2)意味着文件顺利地创建和等待读。然而此文件从路由器不是直接可访问的。请使用CISCO-FTP-CLIENT-MIB读此文件。

使用 CISCO-FTP-CLIENT-MIB 传送文件

对于每FTP客户端操作，您必须选择行实例的一个随机数。您能使用您使用得上述同样随机数的之一。此示例使用555。

逐步指导

使用CISCO-FTP-CLIENT-MIB，要转接文件，请完成这些步骤：

1. 创建FTP客户端的行实例。

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

2. 填写要求的参数。LocalFile必须是名称和您创建上述的文件一样!请使用putASCII转接bulkASCII文件。如果设置cbfDefineFileFormat为以上的bulkBinary，您必须设置cfcRequestOperation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. 通过设置行开始转移到激活。

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

FTP发送开始。当完成时，文件保存对/home/Marcus/ifTable.txt。

4. 要得到FTP发送的状态，再请使用snmpget在cfcRequestResult对象。此对象使用您以其他FTP对象使用的同一个索引。

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

值pending(1)含义文件仍然转接。值success(2)顺利地含义传输的文件。其他值是错误。

5. 当文件是完成的转接时，请再试一次cbfStatusFileState snmpget。它当前有一个不同的值。

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

值emptied(3)意味着文件顺利地读了。文件不可能再转接。

6. 通过毁坏file status行删除此文件当前是安全的。此对象采取索引和上面cbfStatusFileState。

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

7. 一旦文件删除，请删除对应的对象和文件行。

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

照此，使用FTP，您能使用CISCO-FTP-CLIENT-MIB转接所有文件路由器。

验证结果

此部分指南您通过读某些此文件的语法。

1. 第一行是。对于我们的IfTable，它是：

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

这对应于ifEntryOID。ifTable撰写一个或更多ifEntries。

2. 下一条在表里列出对象数量。线路在表里包括对象数量跟随的关键字表，跟随由每个对象索引。例如：

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

表包含22个对象的此线路状态和每个对象有一个增加的索引。这些对象是从IfTable：

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

3. 在此线路以后，有多个行条目。在IfTable，每行对应于接口。行开始与关键字行，跟随由他们的索引标识符和跟随由上一个条目列举的对象。例如：

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

4. 第四个条目是接口的1. ifDescr。然而，这是在十六进制编码的ASCII码的ifDescr。要翻译此线路成更多可读的格式，请使用此perl命令：

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

此条目对应建立接口TokenRing0/0。通常是字符串的所有对象显示作为在大批文件的十六进制编码的ASCII码。您能使用perl命令的此翻译所有十六进制ASCII字符串成可读的文本。如果没有Perl，请使用此[ASCII字符表](#) 翻译字符串。

5. 一些条目显示|值的字符。这意味着该对象的值是NULL。即对象不是在设备的instanciated。例如：

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

这对应于ATM1/0/0-atm层接口。注意ifMtu是此接口的NULL。因为这是虚拟接口，有意义没有MTU。如果prefer，您能用0替换这些空通过添加此命令到设备配置：

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

排除结果中显示的故障

当轮询cbfStatusFileState，如果除running(1)、ready(2)或者emptied(3)收到值，您的操作遇到错误。这些是错误的原因：

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

如果对象数量在文件的比您预计是较少，从CISCO-BULK-FILE-MIB的cbfDefineMaxObjects可能设置的太低。要确定对象的当前值，请使用snmpget。

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

值为0意味着限制没有配置。值可以设置为在0和4294967295之间的任何整数，包括。要设置最大对象每个文件到10，请使用snmpset命令。此对象的索引总是0。

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

此对象可能不是可配置在所有平台。如果snmpset失效与此错误，对象不是可配置在您的平台：

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

当轮询cfcRequestResult对象，如果除pending(1)或success(2)收到值，FTP操作遇到错误。这些是错误的原因：

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

警告

- 目前没有支持的方式直接地访问大批文件。您必须通过CISCO-FTP-CLIENT-MIB读文件。

- `cbfDefineFileStorage`对象定义了三个类型：，和。目前，支持唯一的类型IOS是。短暂文件以直到读的少量存在。
- 一旦文件读，他们不可能再读。必须首先再创他们。
- `cbfDefineFileFormat`对象定义了三个类型：`standardBER`、`bulkBinary`和`bulkASCII`。唯一的支持的格式是`bulkBinary`和`bulkASCII`。默认格式是`bulkBinary`。
- 因为不返回正确结果代码，Windows的Chameleon FTP服务器知道不与CISCO-FTP-CLIENT-MIB一起使用。

[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](#)