

# How To Use the CISCO-BULK-FILE-MIB

## Contents

[Introduction](#)

[Antes de Começar](#)

[Requirements](#)

[Componentes Utilizados](#)

[Conventions](#)

[Informações de Apoio](#)

[Usando o CISCO-BULK-FILE-MIB](#)

[Criando uma operação BILK-FILE](#)

[Step-by-Step Instructions](#)

[Transferindo o arquivo utilizando o CISCO-FTP-CLIENT-MIB](#)

[Step-by-Step Instructions](#)

[Verificando o resultado](#)

[Troubleshooting do Resultado](#)

[Caveats](#)

[Informações Relacionadas](#)

## Introduction

Este documento explica como usar o CISCO-BULK-FILE-MIB e transfere arquivos criados por essa Base de informações de gerenciamento (MIB) usando o CISCO-FTP-CLIENT-MIB.

A partir do software Cisco IOS® versão 12.0, a Cisco implementou uma maneira de armazenar um objeto ou tabela do Protocolo de Gerenciamento de Rede Simples (SNMP - Simple Network Management Protocol) como um arquivo no dispositivo. Esse arquivo pode ser recuperado usando o CISCO-FTP-CLIENT-MIB. Essa tecnologia permite transferir grandes volumes de dados usando um método de transporte confiável.

## Antes de Começar

### Requirements

Antes de tentar esta configuração, verifique se estes requisitos são atendidos:

- Você tem um dispositivo Cisco executando o software Cisco IOS® versão 12.0 ou posterior. Verifique a Ferramenta MIB Locator para certificar-se de que o CISCO-BULK-FILE-MIB seja suportado por seu dispositivo. Um link para a ferramenta pode ser encontrado na página [Cisco IOS MIB Tools](#). **Observação:** esta MIB não é suportada em dispositivos Catalyst OS.
- O SNMP deve ser configurado no dispositivo com as séries de comunidade de somente

leitura e leitura/gravação. Este documento não trata dessa questão. Para obter informações sobre como configurar o SNMP em dispositivos IOS®, leia [Como configurar séries de comunidade SNMP em roteadores, Switches XL baseados em software Cisco IOS, RSMs, MSFCs e Switches Catalyst](#).

## [Componentes Utilizados](#)

As informações neste documento são baseadas nestas versões de software e hardware:

- O CISCO-BULK-FILE-MIB deve armazenar o ifTable de um roteador 7507 executando 12.1(12) em um arquivo, em seguida use o CISCO-FTP-CLIENT-MIB para transferir esse arquivo do roteador para um servidor FTP.
- O conjunto de comandos SNMP [net-snmp](#) é instalado no UNIX ou Windows.
- Esses MIBs são usados:SNMPv2-TCSNMPv2-SMISNMPv2-CONFSNMPv2-MIBIANAifType-MIBIF-MIBCISCO-SMICISCO-TCCISCO-BULK-FILE-MIBCISCO-FTP-CLIENT-MIB

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

## [Conventions](#)

For more information on document conventions, refer to the [Cisco Technical Tips Conventions](#).

## [Informações de Apoio](#)

Verifique se os MIBs nesta tabela estão carregados na sua plataforma de gerenciamento. Isso permite que você use os nomes de objeto e os valores listados acima em vez dos OIDs (identificadores de objeto). Em geral, este documento se refere aos nomes de objeto e não aos OIDs.

Versão 1 Formato SMI	Versão 2 Formato SMI
<a href="#">SNMPv2-SMI-V1SMI.my</a>	<a href="#">SNMPv2-SMI.my</a>
<a href="#">SNMPv2-TC-v1SMI.my</a>	<a href="#">SNMPv2-TC.my</a>
	<a href="#">SNMPv2-CONF.my</a>
<a href="#">SNMPv2-MIB-V1SMI.my</a>	<a href="#">SNMPv2-MIB.my</a>
<a href="#">IANAifType-MIB-V1SMI.my</a>	<a href="#">IANAifType-MIB.my</a>
<a href="#">IF-MIB-V1SMI.my</a>	<a href="#">IF-MIB.my</a>
<a href="#">CISCO-SMI-V1SMI.my</a>	<a href="#">CISCO-SMI.my</a>
<a href="#">CISCO-TC-V1SMI.my</a>	<a href="#">CISCO-TC.my</a>
<a href="#">CISCO-BULK-FILE-MIB-V1SMI.my</a>	<a href="#">CISCO-BULK-FILE-MIB.my</a>
<a href="#">CISCO-FTP-CLIENT-MIB-V1SMI.my</a>	<a href="#">CISCO-FTP-CLIENT-MIB.my</a>

## [Usando o CISCO-BULK-FILE-MIB](#)

## [Criando uma operação BULK-FILE](#)

Neste exemplo, capturamos o `ifTable` de um roteador e o armazenamos em um arquivo em massa. Entretanto, você pode usar qualquer objeto ou tabela MIB.

Use a versão `net-snmp` do `snmpset`. O endereço IP do roteador é 14.32.8.2. A série de comunidade de leitura-gravação é **privada**. A string de comunidade de somente leitura é pública.

Toda vez que você criar uma nova operação de arquivo de grande escala, escolha dois números aleatórios para a instância da fileira. Eles podem ser qualquer número entre 1 e 4294967295, inclusive. Para os fins deste exemplo, use 333 e 444.

## [Step-by-Step Instructions](#)

Para criar uma operação BULK-FILE, faça o seguinte:

1. Configure o arquivo a ser criado.

```
$ 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. Especifique o objeto MIB a ser capturado. Este objeto requer dois índices para a operação correta. O 333 é o 333 da tabela de criação de arquivos acima. 444 é um novo número aleatório usado para o índice principal em `cbfDefineObjectTable`. Este exemplo demonstra o uso de um nome de objeto para `cbfDefineObjectID` (`ifTable`). Você também poderia usar um OID totalmente qualificado aqui.

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

3. Ative as filas recém-criadas. É necessário ter ambos os índices para a linha `cbfDefineObjectTable`.

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

4. Crie o arquivo.

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

O arquivo de grande escala é criado.

5. Verifique se o arquivo foi criado com êxito usando `snmpget` no objeto `cbfStatusFileState`. Este objeto requer dois índices. O primeiro índice é o número aleatório escolhido para a tabela File (333 neste exemplo). O segundo índice depende de quantos arquivos você criou em seu roteador. Como este é seu primeiro arquivo, o índice é 1.

Portanto, use o comando:

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

O valor `running(1)` significa que o arquivo está em processo de ser criado. Um valor de `ready(2)` significa que o arquivo foi criado com êxito e está aguardando para ser lido.

Entretanto, esse arquivo não está diretamente acessível do roteador. Use o `CISCO-FTP-CLIENT-MIB` para ler esse arquivo.

## [Transferindo o arquivo utilizando o CISCO-FTP-CLIENT-MIB](#)

Para cada operação de Cliente FTP, você deve selecionar um número aleatório para a instância de linha. Pode ser utilizado um dos mesmos números aleatórios utilizados acima. Este exemplo usa 555.

## Step-by-Step Instructions

Para transferir o arquivo usando um CISCO-FTP-CLIENT-MIB, faça o seguinte:

1. Crie uma instância de linha do cliente FTP.

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

2. Preencha os parâmetros necessários. O LocalFile deve ter o mesmo nome que o arquivo criado acima! Utilize o comando putASCII para transferir arquivos bulkASCII. Se você definir o cbfDefineFileFormat para o bulkBinary acima, terá que definir o cfcRequestOperation para 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. Comece a transferência definindo a fila a ser ativada.

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

A transferência de FTP é iniciada. Quando concluído, o arquivo será salvo em **/home/Marcus/ifTable.txt**.

4. Para obter o status da transferência de FTP, use **snmpget** novamente no objeto **cfcRequestResult**. Esse objeto utiliza o mesmo índice usado com os outros objetos FTP.

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

Um valor de pending(1) significa que o arquivo ainda está em transferência. Um valor de sucesso (2) significa que o arquivo foi transferido com êxito. Qualquer outro valor é um [erro](#).

5. Quando a transferência do arquivo for concluída, teste o **snmpget** do objeto **cbfStatusFileState** novamente. Agora ele tem um valor diferente.

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

O valor de emptied(3) significa que o arquivo foi lido com sucesso. Não é possível transferir o arquivo novamente.

6. Agora é seguro excluir esse arquivo destruindo sua linha de status. Este objeto tem os mesmos índices que o **cbfStatusFileState** acima.

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

7. Depois que o arquivo é excluído, exclua as filas Object e File correspondentes.

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

Dessa forma, você pode usar o CISCO-FTP-CLIENT-MIB para transferir qualquer arquivo do roteador usando FTP.

## Verificando o resultado

Esta seção orienta você durante a leitura de parte da sintaxe deste arquivo.

1. A primeira linha é a linha de prefixo. Para o nosso exemplo `ifTable`, é:

```
prefix 1.3.6.1.2.1.2.2.1
```

Isso corresponde ao OID do objeto `ifEntry`. A `ifTable` é composta de uma ou mais `ifEntries`.

2. A próxima linha lista o número de objetos da tabela. A linha consiste de uma tabela de palavras-chaves seguida pelo número de objetos na tabela e pelo índice de cada objeto. Por exemplo:

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

Essa linha indica que a tabela contém 22 objetos e cada objeto tem um índice de incremento. Estes objetos são do exemplo `ifTable`:

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

3. Depois dessa linha, há várias entradas de linha. No exemplo de `ifTable`, cada linha corresponde a uma interface. As linhas começam com a linha de palavra-chave, seguida pelo identificador de índice e seguidas pelos objetos enumerados pela entrada da tabela anterior. Por exemplo:

```
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. A quarta entrada é `ifDescr` para a interface 1. No entanto, esse é o `ifDescr` no ASCII hexadecimal criptografado. Para converter esta linha em um formato mais legível, use este comando Perl:

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

Essa entrada corresponde à interface `TokenRing0/0`. Todos os objetos que são normalmente strings são exibidos como ASCII codificado hexadecimal nos arquivos em massa. É possível utilizar este comando Perl para converter qualquer tipo de string ASCII hexadecimal em texto legível. Se você não tem Perl, use esta [tabela](#) de [caracteres ASCII](#) para converter a string.

5. Algumas entradas mostram caracteres `~` para valores. Isso significa que o valor desse objeto é NULL. Ou seja, o objeto não está instanciado no dispositivo. Por exemplo:

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

Isso corresponde à interface da camada `ATM1/0/0-atm`. Observe que `ifMtu` é NULL para esta interface. Como essa é uma interface virtual, faz sentido que ela não tenha um MTU. Se preferir, você pode substituir esses NULLs por `0` adicionando este comando à configuração do dispositivo:

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

## Troubleshooting do Resultado

Ao eleger o objeto `cbfStatusFileState`, se você receber outro valor que não `running(1)` [em execução(1)], `ready(2)` [pronto(2)] ou `emptied(3)` [vazio(3)], isto significará que sua operações encontrou um erro. Estas são as causas dos erros:

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

Se o número de objetos no arquivo for menor do que o esperado, `cbfDefineMaxObjects` do CISCO-BULK-FILE-MIB pode ser definido como muito baixo. Para determinar o valor atual do objeto, use `snmpget`.

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

Um valor 0 significa que nenhum limite está configurado. O valor pode ser definido como qualquer inteiro entre 0 e 4294967295, inclusive. Para definir o máximo de objetos por arquivo como 10, use o comando `snmpset`. O índice para esse objeto é sempre 0.

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

Esse objeto talvez não seja configurável em todas as plataformas. Se o `snmpset` falhar com este erro, o objeto não é configurável na plataforma:

Error in packet.

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

Failed object:

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

Ao executar poll do objeto `cfcRequestResult`, se você receber um valor diferente de pendente(1) ou sucesso(2), a operação de FTP encontrou um erro. Estas são as causas dos erros:

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

## Caveats

- Atualmente não há suporte para acessar os arquivos em massa diretamente. Você deve percorrer o CISCO-FTP-CLIENT-MIB para ler os arquivos.
- O objeto `cbfDefineFileStorage` define três tipos: efêmero, volátil e permanente. Atualmente, o único tipo suportado no IOS é efêmero. Há arquivos efêmeros em pequenas quantidades até a leitura.
- Quando os arquivos são lidos, eles não podem ser lidos novamente. Antes, eles devem ser recriados.

- O objeto `cbfDefineFileFormat` define três tipos: `standardBER`, `bulkBinary` e `bulkASCII`. Os únicos formatos suportados são `bulkBinary` e `bulkASCII`. O formato padrão é `bulkBinary`.
- O servidor FTP Chameleon para Windows é conhecido por não funcionar com o CISCO-FTP-CLIENT-MIB, já que não retorna códigos de resultados corretos.

## Informações Relacionadas

- [Como configurar séries de comunidades do SNMP em roteadores, Switches XL baseados no Cisco IOS Software, RSMs, MSFCs e Switches Catalyst](#)
- [Suporte Técnico - Cisco Systems](#)