

Cómo usar CISCO-BULK-FILE-MIB

Contenido

[Introducción](#)

[Antes de comenzar](#)

[Requirements](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Antecedentes](#)

[Uso de CISCO-BULK-FILE-MIB](#)

[Creación una operación BULK-FILE](#)

[Step-by-Step Instructions](#)

[Transferencia del archivo mediante el uso de CISCO-FTP-CLIENT-MIB](#)

[Step-by-Step Instructions](#)

[Verificación del resultado](#)

[Resolución del resultado](#)

[Advertencias](#)

[Información Relacionada](#)

Introducción

Este documento explica cómo usar la CISCO-BULK-FILE-MIB y cómo transferir archivos creados a partir de dicha Base de información de administración (MIB) mediante la CISCO-FTP-CLIENT-MIB.

A partir de la versión 12.0 del software Cisco IOS®, Cisco ha implementado una forma de almacenar un objeto o una tabla SNMP como un archivo en el dispositivo. Este archivo se puede recuperar utilizando CISCO-FTP-CLIENT-MIB. Esta tecnología le permite transferir grandes cantidades de datos mediante un método de transporte confiable.

Antes de comenzar

Requirements

Antes de utilizar esta configuración, asegúrese de que cumple con estos requisitos:

- Tiene un dispositivo Cisco que ejecuta la versión 12.0 o posterior del software del IOS® de Cisco. Los mensajes de saludo de RGMP son enviados a los mismos intervalos de retransmisión que los mensajes de saludo de PIM (el valor predeterminado es 30 segundos). Se puede encontrar un enlace a la herramienta en la página [Herramientas de MIB de Cisco IOS](#). **Nota:** Esta MIB no se soporta en los dispositivos Catalyst OS.

- SNMP debe estar configurado en el dispositivo con cadenas de comunidad sólo lectura y lectura/escritura. Esto no se trata en este documento. Para obtener información sobre la configuración de SNMP en dispositivos IOS®, lea [Cómo Configurar las Cadenas de Comunidad SNMP en Routers, Switches XL Basados en Software Cisco IOS, RSM, MSFC y Switches Catalyst](#).

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

- El CISCO-BULK-FILE-MIB para almacenar la ifTable desde un router 7507 con 12.1(12) en un archivo, luego use el CISCO-FTP-CLIENT-MIB para transferir ese archivo desde el router a un servidor FTP.
- El conjunto de comandos [net-snmp](#) con SNMP instalado en UNIX o Windows.
- Estas MIB se utilizan: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.

Convenciones

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

Antecedentes

Asegúrese de que los MIB de esta tabla se carguen en su plataforma de administración. Esto le permite utilizar los nombres y valores del objeto enumerados arriba en lugar de utilizar identificadores de objeto (OID) numéricos. En general, este documento se refiere a nombres de objetos y no a OID.

Formato de la versión 1	Formato de la versión 2
SNMPv2-SMI-V1SMI.my	SNMPv2-SMI.my
SNMPv2-TC-v1SMI.my	SNMPv2-TC.my
	SNMPv2-CONF.my
SNMPv2-MIB-V1SMI.my	SNMPv2-SMI.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

Uso de CISCO-BULK-FILE-MIB

Creación una operación BULK-FILE

En este ejemplo, capturamos la `ifTable` de un router y la almacenamos en un archivo masivo. Sin embargo, puede utilizar cualquier objeto o tabla MIB.

Utilice la versión `net-snmp` de `snmpset`. La dirección de IP del router es 14.32.8.2. Su cadena de comunidad de lectura y escritura es **privada**. La cadena de comunidad de sólo lectura es pública.

Cada vez que cree una nueva operación de archivo masivo, elija dos números aleatorios para la instancia de fila. Puede ser cualquier número entre 1 y 4294967295 inclusive. Para los fines de este ejemplo, utilice 333 y 444.

Step-by-Step Instructions

Para crear una operación BULK-FILE, siga estos pasos:

1. Configure el archivo que se va a crear.

```
$ 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 el objeto MIB a capturar. Este objeto requiere dos índices para un funcionamiento correcto. El 333 es el 333 de la tabla de creación de archivos anterior. El 444 es un nuevo número aleatorio utilizado para el índice principal en `cbfDefineObjectTable`. Este ejemplo muestra cómo utilizar un nombre de objeto para `cbfDefineObjectID` (`ifTable`).

También aquí podría utilizar un OID plenamente calificado.

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

3. Activar las filas nuevas creadas. Debe tener ambos índices para la fila de `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. Cree el archivo.

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

Se crea el archivo global.

5. Verifique que el archivo haya sido creado exitosamente mediante la utilización de `snmpget` en el objeto `cbfStatusFileState`. Este objeto requiere dos índices. El primer índice es el número aleatorio elegido para la tabla de archivos (333 en este ejemplo). El segundo índice depende de cuántos archivos creó en su router. Dado que este es su primer archivo, el índice es 1. Por lo tanto, utilice el comando:

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

Un valor de ejecución(1) significa que el archivo está en el proceso de ser creado. Un valor de `ready`(2) significa que el archivo se creó correctamente y está a la espera de ser leído.

No obstante, no se puede acceder a este archivo directamente desde el router. Use el CISCO-FTP-CLIENT-MIB para leer este archivo.

Transferencia del archivo mediante el uso de CISCO-FTP-CLIENT-MIB

Para cada operación de cliente FTP, debe seleccionar un número aleatorio para la instancia de fila. Puede usar uno de los mismos números aleatorios que usó anteriormente. Este ejemplo utiliza 555.

Step-by-Step Instructions

Para transferir el archivo mediante un CISCO-FTP-CLIENT-MIB, siga estos pasos:

1. Cree una instancia de línea del cliente FTP.

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

2. Complete los parámetros requeridos. ¡El archivo LocalFile debe tener el mismo nombre que el archivo creado antes! Use putASCII para transferir los archivos bulkASCII. Si configura cbfDefineFileFormat en bulkBinary arriba, debe configurar cfcRequestOperation en 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. Comience la transferencia al configurar la fila en activa.

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

Se inicia la transferencia FTP. Cuando se completa, el archivo se guarda en **/home/Marcus/ifTable.txt**.

4. Para obtener el estado de la transferencia FTP, utilice **snmpget** de nuevo en el objeto **cfcRequestResult**. Este objeto utiliza el mismo índice que usted utilizó con los otros objetos FTP.

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

Un valor pendiente(1) significa que el archivo aún está transfiriendo. Un valor de acierto(2) significa que el archivo se transfirió exitosamente. Cualquier otro valor es un [error](#).

5. Cuando el archivo ha terminado de transferirse, intente nuevamente el **snmpget** del objeto **cbfStatusFileState**. Tiene ahora un 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)
```

El valor de **emptied(3)** significa que el archivo se leyó satisfactoriamente. El archivo no puede volver a transferirse.

6. Ahora es más seguro eliminar este archivo destruyendo la línea de estado de archivo. Este objeto toma los mismos índices que el **cbfStatusFileState** anterior.

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

7. Una vez que se haya eliminado el archivo, elimine las filas objeto y archivo correspondientes.

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

De esta manera, puede utilizar CISCO-FTP-CLIENT-MIB para transferir cualquier archivo fuera del router mediante FTP.

Verificación del resultado

Esta sección le guía a través de la lectura de parte de la sintaxis de este archivo.

1. La primera línea es la línea de prefijo. Para nuestro ejemplo `ifTable`, es:

```
prefix 1.3.6.1.2.1.2.2.1
```

Esto corresponde al OID para el objeto `ifEntry`. La `ifTable` se compone de una o más `ifEntries`.

2. La siguiente línea detalla el número de objetos en la tabla. La línea consiste en la tabla de palabras clave seguida de la cantidad de objetos en la tabla, seguidos del índice de cada objeto. Por ejemplo:

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

Esta línea indica que la tabla contiene 22 objetos y que cada objeto tiene un índice de aumento. Estos objetos son del ejemplo `ifTable`:

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

3. Luego de esta línea, existen entradas de líneas múltiples. En el ejemplo de `ifTable`, cada fila corresponde a una interfaz. Las filas comienzan por la fila de palabras clave, seguida por su identificador de índice y, a continuación, por los objetos enumerados por la entrada de tabla anterior. Por ejemplo:

```
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. La cuarta entrada es `ifDescr` para la interfaz 1. Sin embargo, éste es el `ifDescr` en ASCII codificado en hexadecimal. Para traducir esta línea a un formato más legible, utilice este comando Perl:

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

Esta entrada corresponde a la interfaz `TokenRing0/0`. Todos los objetos que normalmente son cadenas se muestran como ASCII codificado hexadecimal en los archivos masivos. Puede usar este comando de Perl para convertir cualquier cadena hexadecimal ASCII en texto legible. Si no tiene Perl, utilice esta [tabla](#) de caracteres ASCII para traducir la cadena.

5. Algunas entradas muestran caracteres `~` para los valores. Esto significa que el valor de ese objeto es NULL. Esto es, el objeto no es una instancia en el dispositivo. Por ejemplo:

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

Esto corresponde a la interfaz de capa `ATM1/0/0-atm`. Observe que `ifMtu` es NULL para esta interfaz. Dado que se trata de una interfaz virtual, tiene sentido que no tenga una MTU. Si lo prefiere, puede reemplazar estos NULL con `0` agregando este comando a la configuración del dispositivo:

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

Resolución del resultado

Al consultar el objeto `cbfStatusFileState`, si recibe un valor distinto de ejecución(1), listo(2) o vaciado(3), su operación encontró un error. Estas son las causas de los errores:

```
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
```

Si el número de objetos en el archivo es menor de lo esperado, `cbfDefineMaxObjects` de CISCO-BULK-FILE-MIB puede establecerse en un valor demasiado bajo. Para determinar el valor actual del objeto, utilice `snmpget`.

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

Un valor de 0 significa que no se ha configurado ningún límite. El valor se puede establecer en cualquier entero entre 0 y 4294967295, inclusive. Para establecer el número máximo de objetos por archivo en 10, utilice el comando `snmpset`. El índice de este objeto es siempre 0.

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

Es posible que este objeto no sea configurable en todas las plataformas. Si `snmpset` falla con este error, el objeto no se puede configurar en su 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
```

Cuando se consulta el objeto `cfcRequestResult`, si recibe un valor que no sea `pending(1)` o `success(2)`, se produjo un error en la operación del FTP. Estas son las causas de los errores:

```
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
```

Advertencias

- Actualmente, no hay una forma compatible de acceder directamente a los archivos masivos. Debe pasar por el CISCO-FTP-CLIENT-MIB para leer los archivos.
- El objeto `cbfDefineFileStorage` define tres tipos: efímero, volátil y permanente. Actualmente, el único tipo admitido en IOS es el efímero. Existen pequeñas cantidades de archivos

transitorios hasta que son leídos.

- Una vez leídos los archivos, no se pueden volver a leer. Primero se deben recrear.
- El objeto `cbfDefineFileFormat` define tres tipos: `standardBER`, `bulkBinary` y `bulkASCII`. Los únicos formatos soportados son `Binary` y `bulkASCII`. El formato predeterminado es `bulkBinary`.
- El servidor Chameleon FTP para Windows no suele funcionar con el `CISCO-FTP-CLIENT-MIB` ya que no devuelve códigos de resultado correctos.

[Información Relacionada](#)

- [Cómo configurar las cadenas de comunidad SNMP en routers, switches XL basados en software de Cisco IOS, RSM, MSFC y Catalyst Switches](#)
- [Soporte Técnico - Cisco Systems](#)