

Cómo usar CISCO-BULK-FILE-MIB

Contenido

[Introducción](#)

[Antes de comenzar](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Antecedentes](#)

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

[Creación una operación BULK-FILE](#)

[Instrucciones Paso a Paso](#)

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

[Instrucciones Paso a Paso](#)

[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 del Software Release 12.0 de Cisco IOS®, Cisco ha implementado una manera de salvar un objeto del Simple Network Management Protocol (SNMP) o de presentarlo como archivo en el dispositivo. Este archivo se puede entonces extraer usando el CISCO-FTP-CLIENT-MIB. Esta tecnología le permite transferir grandes cantidades de datos mediante un método de transporte confiable.

Antes de comenzar

Requisitos

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

- Usted tiene un dispositivo de Cisco que funciona con el Software Release 12.0 o Posterior de Cisco IOS®. 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). Un link a la herramienta se puede encontrar en la página de las [herramientas del Cisco IOS MIB](#). **Nota:** Este MIB no es compatible con 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 la información sobre configurar el SNMP en los dispositivos IOS®, leído [cómo configurar las cadenas de comunidad SNMP en el Routers, los XL Switch basado en el Cisco IOS Software, los RS, los MSFC y los switches de 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.
- La habitación del [comando net-snmp snmp](#) instalada en UNIX o Windows.
- Se utiliza este MIB:SNMPv2-TCSNMPv2-SMISNMPv2-CONF SNMPv2-MIBIANAifType-MIBIF-MIBCISCO-SMICISCO-TCCISCO-BULK-FILE-MIBCISCO-FTP-CLIENT-MIB

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando, asegúrese de haber comprendido el impacto que puede tener cualquier comando.

Convenciones

Para obtener más información sobre las convenciones del documento, consulte las [Convenciones de Consejos Técnicos de Cisco](#).

Antecedentes

Asegúrese de tener el MIB en esta tabla cargada 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 S I
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-	CISCO-FTP-CLIENT-

Uso de CISCO-BULK-FILE-MIB

Creación una operación BULK-FILE

En este ejemplo, capturamos el `ifTable` de un router, y lo salvamos en un archivo a granel. Sin embargo, usted puede utilizar cualquier objeto de MIB o tabla.

Utilice la versión red-SNMP del `snmpset`. La dirección de IP del router es 14.32.8.2. Su cadena de comunidad de lectura/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. Con el propósito de este ejemplo, utilice 333 y 444.

Instrucciones Paso a Paso

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

1. Configure el archivo que se creará.

```
$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 la operación correcta. Los 333 es los 333 de la tabla de la creación del archivo arriba. 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. Puesto que éste 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 fue creado con éxito, y está esperando para 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.

Instrucciones Paso a Paso

Para transferir el archivo usando un CISCO-FTP-CLIENT-MIB, complete 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` Transferencia FTP comienza. Cuando es completo, el archivo se guarda a `/home/Marcus/ifTable.txt`.
4. Para conseguir el estatus del transferencia FTP, utilice el `snmpget` otra vez en el objeto del `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` arriba.
`$ 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 este modo, usted puede utilizar el CISCO-FTP-CLIENT-MIB para transferir cualquier archivo apagado del router que usa el FTP.

Verificación del resultado

Esta sección le dirige con la lectura de algo del sintaxis para este archivo.

1. La primera línea es la línea del prefijo. Para nuestro ejemplo de `ifTable`, es:
`prefix 1.3.6.1.2.1.2.2.1` Esto corresponde al OID para el objeto `ifEntry`. El `ifTable` se compone de uno 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 estados que la
tabla contiene 22 objetos, y cada objeto tiene un índice que incrementa. Estos objetos son
del ejemplo de 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 con la fila de la palabra clave, seguida por su identificador del índice, y seguida 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 el ifDescr para la interfaz 1. Sin embargo, éste es el ifDescr en ASCII codificado en hexadecimal. Para traducir esta línea a un más formato legible, utilice este comando perl: `perl -e 'print pack("H*", "546F6B656E52696E67302F30")'` TokenRing0/0 Esta entrada corresponde para interconectar TokenRing0/0. Todos los objetos que son normalmente cadenas se visualizan como ASCII hexadecimal codificado en los archivos a granel. Puede usar este comando de Perl para convertir cualquier cadena hexadecimal ASCII en texto legible. Si usted no tiene Perl, utilice esta [tabla del carácter ASCII](#) para traducir la cadena.

5. Algunas entradas muestran ~ los caracteres para los valores. Esto significa que el valor para ese objeto es NULO. 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 la capa ATM1/0/0-atm. Note que el ifMtu es NULO para esta interfaz. Puesto que esto es una interfaz virtual, tiene sentido que no tiene un MTU. Si usted prefiere, usted puede substituir estos NULL por 0 agregando este comando a la configuración de 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. Éstas son causas para 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 menos que usted espera, los cbfDefineMaxObjects del CISCO-BULK-FILE-MIB se pueden fijar demasiado bajos. Para determinar el valor actual del objeto, utilice el **snmpget**.

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

Un valor de 0 significa que no se configura ningún límite. El valor se puede fijar a cualquier número entero entre 0 y 4294967295, inclusivo. Para fijar los objetos del máximo por el archivo a 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 el **snmpset** falla con este error, el objeto no es configurable 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. Éstas son causas para los errores:

<code>aborted</code>	<code>user aborted the transfer</code>
<code>fileOpenFailLocal</code>	<code>local bulk file was not found</code>
<code>fileOpenFailRemote</code>	<code>remote file could not be opened for writing</code>
<code>badDomainName</code>	<code>FTP server's hostname could not be resolved</code>
<code>unreachableIpAddress</code>	<code>route to the FTP server could not be found</code>
<code>linkFailed</code>	<code>connection could not be made to the remote server</code>
<code>fileReadFailed</code>	<code>local file could not be read</code>
<code>fileWriteFailed</code>	<code>remote file could not be written</code>

Advertencias

- No hay actualmente manera soportada de acceder los archivos a granel directamente. Debe pasar por el `CISCO-FTP-CLIENT-MIB` para leer los archivos.
- El objeto del `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 que se leen los archivos, no pueden ser releídos. Primero se deben recrear.
- El objeto del `cbfDefineFileFormat` define tres tipos: `standardBER`, `bulkBinary`, y `bulkASCII`. Los únicos Formatos admitidos son `bulkBinary` 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](#)