

Hoe gebruikt u de CISCO-BULK-FILE-MIB

Inhoud

[Inleiding](#)

[Voordat u begint](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Conventies](#)

[Achtergrondinformatie](#)

[De CISCO-BULK-FILE-MIB gebruiken](#)

[Een BULKBESTAND-bewerking maken](#)

[Stapsgewijze instructies](#)

[Het bestand overdragen met CISCO-FTP-CLIENT-MIB](#)

[Stapsgewijze instructies](#)

[Het resultaat controleren](#)

[Resultaat oplossen](#)

[Caveats](#)

[Gerelateerde informatie](#)

Inleiding

Dit document legt uit hoe de CISCO-BULK-FILE-MIB en de overdrachtsbestanden moeten worden gebruikt die met die Management Information Base (MIB) zijn gemaakt met behulp van CISCO-FTP-CLIENT-MIB.

Vanaf Cisco IOS® software release 12.0 heeft Cisco een manier geïmplementeerd om een Eenvoudig Network Management Protocol (SNMP) object of tabel op het apparaat op te slaan. Dit bestand kan dan worden opgeroepen met de CISCO-FTP-CLIENT-MIB. Met deze technologie kan je grote hoeveelheden gegevens overdragen met een betrouwbare transportmethode.

Voordat u begint

Vereisten

Zorg er voordat u deze configuratie probeert voor dat u aan deze vereisten voldoet:

- U hebt een Cisco-apparaat dat de Cisco IOS® software release 12.0 of hoger uitvoert. Controleer het MIB Locator Tool om er zeker van te zijn dat CISCO-BULK-FILE-MIB voor uw apparaat wordt ondersteund. Een link naar het gereedschap is te vinden op de pagina [Cisco IOS MIB-tools](#). **Opmerking:** deze MIB wordt niet ondersteund op Catalyst OS-apparaten.
- SNMP moet op het apparaat worden geconfigureerd met zowel alleen-lezen als

gemeenschapskoorden die lezen. Dit wordt niet in dit document behandeld. Voor informatie bij het configureren van SNMP op IOS® apparaten, lees [hoe u SNMP Community Strings op Routers kunt configureren, Cisco IOS op software gebaseerde XL Switches, RSM's, MSFC's en Catalyst Switches](#).

Gebruikte componenten

De informatie in dit document is gebaseerd op de volgende software- en hardware-versies:

- CISCO-BULK-FILE-MIB om de ifTable op te slaan van een 7507 router die 12.1(12) in een bestand draait. Gebruik vervolgens CISCO-FTP-CLIENT-MIB om dat bestand van de router naar een FTP-server over te brengen.
- De [net-snmp](#) SNMP commandosuite geïnstalleerd op UNIX of Windows.
- Deze MIB's worden gebruikt:SNMPv2-TCSNMPv2-SMISNMPv2-CONF SNMPv2-MIBIANAifType-MIBINDIEN-MIBCISCO-SMICISCO-TCCISCO-BULK-FILE-MIBCISCO-FTP-CLIENT-MIB

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u de potentiële impact van elke opdracht begrijpen.

Conventies

Raadpleeg [Cisco Technical Tips Conventions](#) (Conventies voor technische tips van Cisco) voor meer informatie over documentconventies.

Achtergrondinformatie

Zorg ervoor dat de MIB's in deze tabel in uw beheerplatform zijn geladen. Dit staat u toe om de objectnamen en -waarden hierboven te gebruiken in plaats van de numerieke Objectidentificatoren (OIDs). In het algemeen verwijst dit document naar objectnamen en niet naar OID's.

Versie 1 SMI-indeling	Versie 2 SMI-indeling
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

De CISCO-BULK-FILE-MIB gebruiken

Een BULKBESTAND-bewerking maken

In dit voorbeeld vangen we de `ifTable` van een router en slaan het op in een bulkbestand. U kunt echter elk object of elke tabel van MIB gebruiken.

Gebruik de netto-versie van de **computer**. Het IP-adres van de router is **14.32.8.2**. Zijn read-writer string is **privé**. De alleen-lezen community string is **openbaar**.

Telkens wanneer u een nieuwe bulkbestandhandeling maakt, kiest u twee willekeurige getallen voor de rijinstantie. Ze kunnen elk getal zijn tussen 1 en 4294967295 inclusief. Gebruik voor dit voorbeeld 333 en 444.

Stapsgewijze instructies

Voltooi de volgende stappen om een BULK-FILE-bewerking te maken:

1. Stel het te maken bestand in.

```
$ 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. Specificeer het MIB-object dat moet worden opgenomen. Dit object heeft twee indexen nodig voor een correct gebruik. 333 is de 333 uit de tabel voor het maken van bestanden hierboven. De 444 is een nieuw willekeurig aantal dat voor de primaire index in de `cbfDefineObjectTable` wordt gebruikt. Dit voorbeeld demonstreert het gebruiken van een objectnaam voor `cbfDefineObjectID (ifTable)`. Je zou hier ook een volledig gekwalificeerde OID kunnen gebruiken.

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

3. Activeert de nieuwe rijen. U moet beide indexen hebben voor uw `cbfDefineObjectTable` rij.

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

4. Maak het bestand.

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

Het bulkbestand wordt gemaakt.

5. Controleer of het bestand met succes is gemaakt met behulp van de `snmpget` in het `cbfStatusFileState`-object. Dit object vereist twee indices. De eerste index is het willekeurige nummer dat is geselecteerd voor de Bestandstabel (333 in dit voorbeeld). De tweede index hangt af van het aantal bestanden dat u in uw router hebt gemaakt. Aangezien dit uw eerste bestand is, is de index 1. Gebruik daarom de opdracht:

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

Een waarde van `actief(1)` betekent dat het bestand momenteel wordt aangemaakt. Een waarde van `gereed(2)` betekent dat het bestand met succes is gemaakt en wacht op het lezen.

Dit bestand is echter niet rechtstreeks toegankelijk vanaf de router. Gebruik CISCO-FTP-CLIENT-

MIB om dit bestand te lezen.

Het bestand overdragen met CISCO-FTP-CLIENT-MIB

Bij elke FTP-client moet u een willekeurig nummer voor de rijinstantie selecteren. U kunt één van de zelfde willekeurige nummers gebruiken die u hierboven gebruikte. Dit voorbeeld gebruikt 555.

Stapsgewijze instructies

Als u het bestand wilt overdragen met een CISCO-FTP-CLIENT-MIB, voert u deze stappen uit:

1. Maak een rij-exemplaar van de FTP-client.

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

2. Vul de gewenste parameters in. Het LocalFile **moet** dezelfde naam zijn als het bestand dat u hierboven hebt gemaakt! Gebruik **putASCII** om **bulkASCII** bestanden over te brengen. Als u het **cbfDefineFileFormat** instelt op **bulkBinary** hierboven, moet u **cfcApplication Operatie** instellen **OM** Binair te maken.

```
$ 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. Begin met de overdracht door de rij in te stellen op actief.

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

De FTP-overdracht begint. Na voltooiing, wordt het bestand opgeslagen op **/home/Marcus/ifTable.txt**.

4. Om de status van de FTP-overdracht te krijgen, **gebruikt u de snmpget** opnieuw op het **cfcrequestResult**-object. Dit object gebruikt dezelfde index als bij de andere FTP-objecten.

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

Een waarde van **hangende(1)** betekent dat het bestand nog steeds overdraagt. Een waarde van **succes(2)** betekent het overgedragen bestand. Elke andere waarde is een **fout**.

5. Wanneer het bestand overgemaakt is, probeert u het **pictogram** van het **cbfStatusFileState** opnieuw. Het heeft nu een andere waarde.

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

De waarde van **geleegd(3)** betekent dat het bestand is gelezen. Het bestand kan niet opnieuw worden overgedragen.

6. Het is nu veilig om dit bestand te verwijderen door de rij voor de bestandstatus te vernietigen. Dit object krijgt dezelfde indices als de bovenstaande **cbfStatusFileState**.

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

7. Verwijdert het bestand zodra het is verwijderd, het betreffende object en de rijen van het bestand.

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

Op deze manier kunt u CISCO-FTP-CLIENT-MIB gebruiken om een bestand van de router over te dragen met behulp van FTP.

Het resultaat controleren

Deze sectie helpt u bij het lezen van een aantal syntax voor dit bestand.

1. De eerste regel is de regel `prefix`. Voor ons `ifTable` voorbeeld is het:

```
prefix 1.3.6.1.2.1.2.2.1
```

Dit komt overeen met de OID voor het `ifEntry` object. De `ifTable` bestaat uit een of meer `if`.

2. De volgende regel toont het aantal objecten in de tabel. De lijn bestaat uit de sleutelwoordenlijst gevolgd door het aantal objecten in de tabel, gevolgd door de index van elk object. Bijvoorbeeld:

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

Deze lijn verklaart dat de tabel 22 objecten bevat, en elk object heeft een stijgende index.

Deze objecten komen uit het `ifTable` voorbeeld:

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

3. Na deze regel worden er meerdere rijen items ingevoegd. In het voorbeeld `ifTable` correspondeert elke rij met een interface. De rijen beginnen met de sleutelwoordenrij, gevolgd door hun indexidentificatie, en gevolgd door de objecten die door de vorige tabelingang zijn opgesomd. Bijvoorbeeld:

```
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. De vierde ingang is de `ifDescr` voor interface 1. Dit is echter de `ifDescr` in hexadecimale gecodeerde ASCII. U kunt deze regel als volgt vertalen in een leesbaarder indeling: gebruik deze opdracht:

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

Deze ingang komt overeen met interface `TokenRing0/0`. Alle objecten die normaal strijken zijn worden weergegeven als hexadecimale gecodeerde ASCII in de bulkbestanden. U kunt deze opdracht Perl gebruiken om een hexadecimale ASCII-string te vertalen in leesbare tekst. Als u geen Perl hebt, gebruik dan deze [ASCII tekentabel](#) om de string te vertalen.

5. Sommige items tonen ~ karakters voor waarden. Dit betekent dat de waarde voor dat object NULL is. Dat wil zeggen, het object is niet op het apparaat geïnstalleerd. Bijvoorbeeld:

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

Dit komt overeen met de `ATM1/0/0` ATM laaginterface. Merk op dat `alsMtu` NULL is voor deze interface. Aangezien het om een virtuele interface gaat, is het logisch dat deze geen MTU heeft. Als u dit wenst, kunt u deze NULLs met `0` vervangen door deze opdracht aan de configuratie van het apparaat toe te voegen:

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

Resultaat oplossen

Bij het opvragen van het `cbfStatusFileState`-object is uw operatie een fout opgetreden als u een andere waarde ontvangt dan `actief(1)`, `klaar(2)` of `geleegd(3)`. Dit zijn oorzaken voor de fouten:

```
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
```

Als het aantal objecten in het bestand kleiner is dan u verwacht, is mogelijk dat `cbfDefineMaxObjects` van CISCO-BULK-FILE-MIB te laag is ingesteld. Om de huidige waarde van het object te bepalen, gebruikt u een bom.

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

Een waarde van 0 betekent dat er geen limiet wordt ingesteld. De waarde kan worden ingesteld op elk integer tussen 0 en 4294967295, inclusief. Als u de maximale objecten per bestand op 10 wilt instellen, gebruikt u de opdracht **SNMP**. De index voor dit object is altijd 0.

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

Dit object kan niet op alle platforms instelbaar zijn. Als de bom faalt met deze fout, is het object niet Configureerbaar op uw platform:

```
Error in packet.
```

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

```
Failed object:
```

```
enterprises.cisco.ciscoMgmt.ciscoBulkFileMIB.ciscoBulkFileMIBObjects.cbfDefine.cbfDefineMaxObje
ts.0
```

Bij het opiniepeilen van het `cfcApplicationResultaat`-object en als u een andere waarde dan `hangend(1)` of `succes(2)` ontvangt, heeft de FTP-operatie een fout aangetroffen. Dit zijn oorzaken voor de fouten:

```
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

- Op dit moment is er geen ondersteunde manier om de bulkbestanden rechtstreeks te benaderen. U moet door de CISCO-FTP-CLIENT-MIB gaan om de bestanden te lezen.
- Het object `cbfDefineFileStorage` definieert drie typen: vluchtig, vluchtig en permanent. Op dit moment is het enige type dat in IOS wordt ondersteund `kortstondig`. Ephemeral bestanden bestaan in kleine hoeveelheden tot ze gelezen zijn.
- Nadat de bestanden zijn gelezen, kunnen ze niet meer worden gelezen. Ze moeten eerst

opnieuw gecreëerd worden.

- Het object `cbfDefineFileFormat` definieert drie typen: `StandardBER`, `bulkBinary`, en `bulkASCII`. De enige ondersteunde formaten zijn `bulkBinary` en `bulkASCII`. Het standaardformaat is `bulkBinary`.
- Het is bekend dat de FTP-server van Chameleon voor Windows **niet** met CISCO-FTP-CLIENT-MIB kan werken omdat deze geen correcte resultaatcodes teruggeeft.

Gerelateerde informatie

- [Hoe te om SNMP Community-Strings op routers, Cisco IOS-softwaregebaseerde XL-Switches, RSM's, MSFC's en Catalyst-Switches te configureren](#)
- [Technische ondersteuning - Cisco-systemen](#)