

# Verwendung der CISCO-BULK-FILE-MIB

## Inhalt

[Einführung](#)

[Bevor Sie beginnen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Konventionen](#)

[Hintergrundinformationen](#)

[Verwenden der CISCO-BULK-FILE-MIB](#)

[Erstellen einer BULK-DATEI-Operation](#)

[Schrittweise Anleitung](#)

[Übertragen der Datei mithilfe der CISCO-FTP-CLIENT-MIB](#)

[Schrittweise Anleitung](#)

[Überprüfen des Ergebnisses](#)

[Fehlerbehebung am Ergebnis](#)

[Einsprüche](#)

[Zugehörige Informationen](#)

## Einführung

In diesem Dokument wird erläutert, wie Sie die CISCO-BULK-FILE-MIB verwenden und Dateien übertragen, die von dieser Management Information Base (MIB) mithilfe der CISCO-FTP-CLIENT-MIB erstellt wurden.

Ab der Cisco IOS® Softwareversion 12.0 hat Cisco eine Möglichkeit implementiert, ein Simple Network Management Protocol (SNMP)-Objekt oder eine SNMP-Tabelle als Datei auf dem Gerät zu speichern. Diese Datei kann dann mit der CISCO-FTP-CLIENT-MIB abgerufen werden. Mit dieser Technologie können Sie große Datenmengen mithilfe einer zuverlässigen Transportmethode übertragen.

## Bevor Sie beginnen

### Anforderungen

Stellen Sie vor dem Versuch dieser Konfiguration sicher, dass Sie die folgenden Anforderungen erfüllen:

- Auf einem Cisco Gerät wird die Cisco IOS® Software 12.0 oder höher ausgeführt. Überprüfen Sie das MIB Locator Tool, um sicherzustellen, dass die CISCO-BULK-FILE-MIB für Ihr Gerät unterstützt wird. Einen Link zum Tool finden Sie auf der Seite [Cisco IOS MIB Tools](#). **Hinweis:** Diese MIB wird auf Catalyst OS-Geräten nicht unterstützt.

- SNMP muss auf dem Gerät mit schreibgeschützten und schreibgeschützten Community-Strings konfiguriert werden. Dies wird in diesem Dokument nicht behandelt. Informationen zur Konfiguration von SNMP auf IOS®-Geräten finden Sie im Dokument [Konfigurieren von SNMP-Community-Strings auf Routern, Cisco IOS Software-basierten XL-Switches, RSMs, MSFCs und Catalyst Switches](#).

## Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf den folgenden Software- und Hardwareversionen:

- Die CISCO-BULK-FILE-MIB speichert die ifTable von einem 7507-Router mit 12.1(12) in einer Datei und überträgt diese Datei dann mithilfe der CISCO-FTP-CLIENT-MIB vom Router auf einen FTP-Server.
- Die [net-snmp](#)  SNMP-Befehlssuite, die unter UNIX oder Windows installiert ist.
- Diese MIBs werden verwendet:SNMPv2-TCSNMPv2-SMISNMPv2-CONFSNMPv2-MIBIANAifType-MIBIF-MIBCISCO SMICISCO-TCCISCO-BULK-DATEI-MIBCISCO-FTP-CLIENT-MIB

Die Informationen in diesem Dokument wurden von den Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

## Konventionen

Weitere Informationen zu Dokumentkonventionen finden Sie in den [Cisco Technical Tips Conventions](#) .

## Hintergrundinformationen

Stellen Sie sicher, dass die MIBs in dieser Tabelle in die Verwaltungsplattform geladen sind. Auf diese Weise können Sie anstelle der numerischen OIDs (Object Identifiers) die oben aufgeführten Objektnamen und Werte verwenden. Im Allgemeinen bezieht sich dieses Dokument auf Objektnamen und nicht auf OIDs.

SMI-Format Version 1	SMI-Format Version 2
<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-</a>	<a href="#">CISCO-FTP-CLIENT-</a>

## Verwenden der CISCO-BULK-FILE-MIB

### Erstellen einer BULK-DATEI-Operation

In diesem Beispiel erfassen wir die `ifTable` von einem Router und speichern sie in einer Massendatei. Sie können jedoch jedes MIB-Objekt oder jede MIB-Tabelle verwenden.

Verwenden Sie die net-snmp-Version von **snmpset**. Die IP-Adresse des Routers lautet **14.32.8.2**. Der Community-String für Lese- und Schreibvorgänge ist **privat**. Der schreibgeschützte Community-String ist **öffentlich**.

Wählen Sie jedes Mal, wenn Sie eine neue Massendateioperation erstellen, zwei Zufallszahlen für die Zeileninstanz aus. Es kann sich um eine beliebige Zahl zwischen 1 und 4294967295 handeln. Verwenden Sie für dieses Beispiel 333 und 444.

### Schrittweise Anleitung

Gehen Sie wie folgt vor, um einen BULK-FILE-Vorgang zu erstellen:

1. Richten Sie die zu erstellende Datei ein.

```
$ 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. Geben Sie das zu erfassende MIB-Objekt an. Dieses Objekt benötigt zwei Indizes für die korrekte Operation. Der 333 ist der 333 aus der oben stehenden Tabelle zur Dateierstellung. Der 444 ist eine neue Zufallszahl, die für den primären Index in der `cbfDefineObjectTable` verwendet wird. In diesem Beispiel wird die Verwendung eines Objektnamens für `cbfDefineObjectID (ifTable)` veranschaulicht. Sie können hier auch eine vollqualifizierte OID verwenden.

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

3. Aktivieren Sie die neu erstellten Zeilen. Sie müssen über beide Indizes für die `cbfDefineObjectTable`-Zeile verfügen.

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

4. Erstellen Sie die Datei.

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

Die Massendatei wird erstellt.

5. Überprüfen Sie, ob die Datei mithilfe von **snmpget** im `cbfStatusFileState`-Objekt erfolgreich erstellt wurde. Dieses Objekt erfordert zwei Indizes. Der erste Index ist die Zufallszahl, die für die Dateitabelle gewählt wurde (in diesem Beispiel 333). Der zweite Index hängt von der Anzahl der Dateien ab, die Sie auf Ihrem Router erstellt haben. Da dies Ihre erste Datei ist, ist der Index 1. Verwenden Sie daher den Befehl:

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

Ein Wert von `running(1)` bedeutet, dass die Datei gerade erstellt wird. Ein Wert von `ready(2)` bedeutet, dass die Datei erfolgreich erstellt wurde und darauf wartet, gelesen zu werden. Diese Datei ist jedoch nicht direkt vom Router aus zugänglich. Verwenden Sie `CISCO-FTP-CLIENT-MIB`, um diese Datei zu lesen.

## Übertragen der Datei mithilfe der CISCO-FTP-CLIENT-MIB

Für jeden FTP-Client-Vorgang müssen Sie eine Zufallszahl für die Zeileninstanz auswählen. Sie können eine der gleichen Zufallszahlen verwenden, die Sie oben verwendet haben. In diesem Beispiel wird 555 verwendet.

### Schrittweise Anleitung

Gehen Sie wie folgt vor, um die Datei mithilfe einer `CISCO-FTP-CLIENT-MIB` zu übertragen:

1. Erstellen Sie eine Zeileninstanz des FTP-Clients.

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

2. Geben Sie die erforderlichen Parameter ein. Die lokale Datei **muss** mit der oben erstellten Datei identisch sein! Verwenden Sie `putASCII`, um `BulkASCII`-Dateien zu übertragen. Wenn Sie das `cbfDefineFileFormat` auf die obige `BulkBinary` festlegen, müssen Sie

`cfcRequestOperation` auf `putBinary` festlegen.

```
$ 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. Beginnen Sie die Übertragung, indem Sie die Zeile auf "active" setzen.

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

Die FTP-Übertragung wird gestartet. Nach Abschluss dieses Vorgangs wird die Datei unter `/home/Marcus/ifTable.txt` gespeichert.

4. Um den Status des FTP-Übertrags abzurufen, verwenden Sie `snmpget` erneut für das `cfcRequestResult`-Objekt. Dieses Objekt verwendet den gleichen Index, den Sie mit den anderen FTP-Objekten verwendet haben.

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

Ein Wert von `ausstehend(1)` bedeutet, dass die Datei noch immer übertragen wird. Ein Wert von `Success(2)` bedeutet, dass die Datei erfolgreich übertragen wurde. Jeder andere Wert ist ein [Fehler](#).

5. Wenn die Dateiübertragung abgeschlossen ist, versuchen Sie das `snmpget` des `cbfStatusFileState`-Objekts erneut. Jetzt hat es einen anderen Wert.

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

Der Wert von `leerded(3)` bedeutet, dass die Datei erfolgreich gelesen wurde. Die Datei kann nicht erneut übertragen werden.

6. Es ist jetzt sicher, diese Datei zu löschen, indem die Dateistatuszeile zerstört wird. Dieses Objekt verwendet die gleichen Indizes wie der obige `cbfStatusFileState`.

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

7. Löschen Sie nach dem Löschen der Datei die entsprechenden Object- und File-Zeilen.

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

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

Auf diese Weise können Sie jede beliebige Datei des Routers über FTP mit der CISCO-FTP-CLIENT-MIB übertragen.

## Überprüfen des Ergebnisses

Dieser Abschnitt führt Sie durch das Lesen einiger Syntax für diese Datei.

1. Die erste Zeile ist die Präfixzeile. Für das `ifTable`-Beispiel ist dies:

```
prefix 1.3.6.1.2.1.2.2.1
```

Dies entspricht der OID für das `ifEntry`-Objekt. Die `ifTable` besteht aus einem oder mehreren `ifEntries`.

2. In der nächsten Zeile wird die Anzahl der Objekte in der Tabelle aufgelistet. Die Zeile besteht aus der Schlüsselwortabelle gefolgt von der Anzahl der Objekte in der Tabelle, gefolgt vom Index der einzelnen Objekte. Beispiel:

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

Diese Zeile gibt an, dass die Tabelle 22 Objekte enthält, und jedes Objekt verfügt über einen inkrementellen Index. Diese Objekte stammen aus dem Beispiel `ifTable`:

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

3. Nach dieser Zeile gibt es mehrere Zeileneinträge. Im Beispiel `ifTable` entspricht jede Zeile einer Schnittstelle. Die Zeilen beginnen mit der Schlüsselwortzeile, gefolgt von ihrem Indexbezeichner, gefolgt von den Objekten, die durch den vorherigen Tabelleneintrag aufgelistet werden. Beispiel:

```
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. Der vierte Eintrag ist der `ifDescr` für Schnittstelle 1. Dies ist jedoch der `ifDescr` in hexadezimal codierten ASCII. Um diese Zeile in ein lesbareres Format zu übersetzen, verwenden Sie den folgenden Perl-Befehl:

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

```
TokenRing0/0
```

Dieser Eintrag entspricht der Schnittstelle `TokenRing0/0`. Alle Objekte, die normalerweise Zeichenfolgen sind, werden als hexadezimal codierte ASCII in den Massendateien angezeigt. Sie können diesen Perl-Befehl verwenden, um jede hexadezimale ASCII-Zeichenfolge in lesbaren Text zu übersetzen. Wenn Sie Perl nicht haben, verwenden Sie diese [ASCII-Zeichentabelle](#) , um die Zeichenfolge zu übersetzen.

5. Einige Einträge zeigen ~ Zeichen für Werte an. Das bedeutet, dass der Wert für dieses Objekt NULL ist. Das heißt, das Objekt wird nicht auf dem Gerät instanziiert. Beispiel:

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

Dies entspricht der `ATM1/0/0-atm-Layer`-Schnittstelle. Beachten Sie, dass `ifMtu` für diese Schnittstelle NULL ist. Da es sich um eine virtuelle Schnittstelle handelt, ist es sinnvoll, dass diese keine MTU aufweist. Wenn Sie es vorziehen, können Sie diese NULLs durch 0 ersetzen, indem Sie diesen Befehl zur Gerätekonfiguration hinzufügen:

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

## Fehlerbehebung am Ergebnis

Wenn Sie das `cbfStatusFileState`-Objekt abfragen, wenn Sie einen anderen Wert als `running(1)`, `ready(2)` oder `leohne(3)` erhalten, ist bei der Operation ein Fehler aufgetreten. Dies sind die Fehlerursachen:

```
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
```

Wenn die Anzahl der Objekte in der Datei geringer ist als erwartet, kann `cbfDefinitionMaxObjects` aus der CISCO-BULK-FILE-MIB zu niedrig eingestellt werden. Um den aktuellen Wert des Objekts zu bestimmen, verwenden Sie **snmpget**.

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

Ein Wert von 0 bedeutet, dass kein Grenzwert konfiguriert ist. Der Wert kann auf eine beliebige Ganzzahl zwischen 0 und einschließlich 4294967295 festgelegt werden. Um die maximalen Objekte pro Datei auf 10 festzulegen, verwenden Sie den Befehl **snmpset**. Der Index für dieses Objekt ist immer 0.

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

Dieses Objekt kann möglicherweise nicht auf allen Plattformen konfiguriert werden. Wenn **snmpset** mit diesem Fehler fehlschlägt, ist das Objekt auf Ihrer Plattform nicht konfigurierbar:

```
Error in packet.
```

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

```
Failed object:
```

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

Wenn Sie das `cfcRequestResult`-Objekt abfragen, wenn Sie einen anderen Wert als `ausstehend(1)` oder `Success(2)` erhalten, ist beim FTP-Vorgang ein Fehler aufgetreten. Dies sind die Fehlerursachen:

```
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
```

## Einsprüche

- Derzeit wird kein direkter Zugriff auf die Bulk-Dateien unterstützt. Sie müssen die CISCO-FTP-CLIENT-MIB durchlaufen, um die Dateien zu lesen.
- Das `cbfDefineFileStorage`-Objekt definiert drei Typen: `kurzlebig`, `flüchtig` und `dauerhaft`. Derzeit wird in IOS nur der einzige Typ unterstützt, der `kurzlebig` ist. Ephemere Dateien existieren bis zum Lesen in kleinen Mengen.
- Wenn die Dateien gelesen wurden, können sie nicht erneut gelesen werden. Sie müssen zunächst neu erstellt werden.
- Das `cbfDefinitionFileFormat`-Objekt definiert drei Typen: `StandardBER`, `BulkBinary` und `BulkASCII`. Die einzigen unterstützten Formate sind `BulkBinary` und `BulkASCII`. Das Standardformat ist `BulkBinary`.
- Es ist bekannt, dass der Chameleon FTP-Server für Windows **nicht** mit der CISCO-FTP-CLIENT-MIB funktioniert, da er keine korrekten Ergebniscodes zurückgibt.

## Zugehörige Informationen

- [Konfigurieren von SNMP-Community-Strings auf Routern, Cisco IOS Software-basierten XL-Switches, RSMs, MSFCs und Catalyst Switches](#)
- [Technischer Support - Cisco Systems](#)