

Come utilizzare il CISCO-BULK-FILE-MIB

Sommario

[Introduzione](#)

[Operazioni preliminari](#)

[Requisiti](#)

[Componenti usati](#)

[Convenzioni](#)

[Premesse](#)

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

[Creazione di un'operazione BULK-FILE](#)

[Istruzioni dettagliate](#)

[Trasferimento del file con CISCO-FTP-CLIENT-MIB](#)

[Istruzioni dettagliate](#)

[Verifica del risultato](#)

[Risoluzione dei problemi relativi al risultato](#)

[Avvertenze](#)

[Informazioni correlate](#)

Introduzione

Questo documento spiega come utilizzare CISCO-BULK-FILE-MIB e trasferire i file creati da tale Management Information Base (MIB) utilizzando CISCO-FTP-CLIENT-MIB.

A partire dal software Cisco IOS® versione 12.0, Cisco ha implementato un modo per memorizzare un oggetto o una tabella SNMP (Simple Network Management Protocol) come file sul dispositivo. È quindi possibile recuperare il file con CISCO-FTP-CLIENT-MIB. Questa tecnologia consente di trasferire grandi quantità di dati utilizzando un metodo di trasporto affidabile.

Operazioni preliminari

Requisiti

Prima di provare la configurazione, verificare che siano soddisfatti i seguenti requisiti:

- Su un dispositivo Cisco è in esecuzione il software Cisco IOS® versione 12.0 o successive. Controllare lo strumento MIB Locator per verificare che CISCO-BULK-FILE-MIB sia supportato per il dispositivo. Un collegamento allo strumento è disponibile nella pagina [degli strumenti MIB di Cisco IOS](#). **Nota:** questo MIB non è supportato sui dispositivi Catalyst OS.
- Il protocollo SNMP deve essere configurato sul dispositivo con le stringhe della community di

sola lettura e di lettura/scrittura. Argomento non trattato nel presente documento. Per informazioni sulla configurazione del protocollo SNMP sui dispositivi IOS®, consultare il documento sulla [configurazione delle stringhe della community SNMP su router, switch XL, RSM, MSFC e switch Catalyst basati su software Cisco IOS](#).

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

- CISCO-BULK-FILE-MIB per archiviare l'ifTable da un router 7507 con versione 12.1(12) in un file, quindi usare CISCO-FTP-CLIENT-MIB per trasferire il file dal router a un server FTP.
- La suite di comandi [net-snmp](#) e SNMP installata in UNIX o Windows.
- I MIB utilizzati sono:SNMPv2-TCSNMPv2-SMISNMPv2-CONF SNMPv2-MIBIANAifType-MIBIF-MIBCISCO-SMICISCO-TCCISCO-BULK-FILE-MIBCISCO-FTP-CLIENT-MIB

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Convenzioni

Per ulteriori informazioni sulle convenzioni usate, consultare il documento [Cisco sulle convenzioni nei suggerimenti tecnici](#).

Premesse

Accertarsi di avere caricato i MIB in questa tabella nella piattaforma di gestione. In questo modo è possibile utilizzare i nomi e i valori degli oggetti elencati in precedenza anziché gli identificatori di oggetto (OID) numerici. In generale, questo documento fa riferimento ai nomi degli oggetti e non agli OID.

Formato SMI versione 1	Formato SMI versione 2
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

Uso di CISCO-BULK-FILE-MIB

Creazione di un'operazione BULK-FILE

Nell'esempio, l'oggetto `ifTable` viene acquisito da un router e memorizzato in un file bulk. Tuttavia, è possibile utilizzare qualsiasi oggetto o tabella MIB.

Usare la versione `net-snmp` di `snmpset`. L'indirizzo IP del router è **14.32.8.2**. La stringa della community di lettura/scrittura è **privata**. La stringa della community di sola lettura è **pubblica**.

Ogni volta che si crea una nuova operazione per i file in blocco, scegliere due numeri casuali per l'istanza di riga. Possono essere numeri compresi tra 1 e 4294967295 inclusi. Per questo esempio, utilizzare 333 e 444.

Istruzioni dettagliate

Per creare un'operazione BULK-FILE, attenersi alla seguente procedura:

1. Impostare il file da creare.

```
$ 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. Specificare l'oggetto MIB da acquisire. L'oggetto richiede due indici per il corretto funzionamento. Il 333 è il 333 della tabella di creazione dei file riportata sopra. Il valore 444 è un nuovo numero casuale utilizzato per l'indice primario in `cbfDefineObjectTable`. In questo esempio viene illustrato l'utilizzo di un nome di oggetto per `cbfDefineObjectID` (`ifTable`). È possibile utilizzare anche un OID completo.

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

3. Attivate le nuove righe create. È necessario disporre di entrambi gli indici per la riga

```
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. Creare il file.

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

Viene creato il file ausiliario.

5. Verificare che il file sia stato creato correttamente utilizzando `snmpget` sull'oggetto

```
cbfStatusFileState. Questo oggetto richiede due indici. Il primo indice è il numero casuale scelto per la tabella File (333 in questo esempio). Il secondo indice dipende dal numero di file creati nel router. Poiché si tratta del primo file, l'indice è 1. Utilizzare quindi il comando:
```

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

Il valore `running(1)` indica che il file è in fase di creazione. Il valore `ready(2)` indica che il file è stato creato correttamente ed è in attesa di essere letto.

Tuttavia, questo file non è accessibile direttamente dal router. Utilizzare `CISCO-FTP-CLIENT-MIB` per leggere questo file.

Trasferimento del file con CISCO-FTP-CLIENT-MIB

Per ogni operazione del client FTP, è necessario selezionare un numero casuale per l'istanza della riga. È possibile utilizzare uno degli stessi numeri casuali utilizzati in precedenza. In questo esempio viene utilizzato 555.

Istruzioni dettagliate

Per trasferire il file con un CISCO-FTP-CLIENT-MIB, attenersi alla seguente procedura:

1. Creare un'istanza di riga del client FTP.

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

2. Immettere i parametri richiesti. LocalFile **deve** avere lo stesso nome del file creato in precedenza. Utilizzare **putASCII** per trasferire file BulkASCII. Se si imposta `cbfDefineFileFormat` su `bulkBinary`, è necessario impostare `cfcRequestOperation` **SU** `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. Iniziare il trasferimento impostando la riga su attiva.

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

Il trasferimento FTP ha inizio. Al termine, il file viene salvato in `/home/Marcus/ifTable.txt`.

4. Per ottenere lo stato del trasferimento FTP, utilizzare nuovamente **snmpget** sull'oggetto `cfcRequestResult`. Questo oggetto utilizza lo stesso indice utilizzato con gli altri oggetti FTP.

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

Il valore `pending(1)` indica che il file è ancora in fase di trasferimento. Il valore `success(2)` indica che il file è stato trasferito correttamente. Qualsiasi altro valore è un [errore](#).

5. Al termine del trasferimento del file, provare a eseguire di nuovo lo **snmpget** dell'oggetto `cbfStatusFileState`. Ora ha un valore diverso.

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

Il valore di `svuotato(3)` indica che il file è stato letto correttamente. Impossibile trasferire di nuovo il file.

6. È ora possibile eliminare il file eliminando la riga di stato del file. L'oggetto utilizza gli stessi indici dell'oggetto `cbfStatusFileState` precedente.

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

7. Una volta eliminato il file, eliminare le righe Object e File corrispondenti.

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

In questo modo, è possibile utilizzare CISCO-FTP-CLIENT-MIB per trasferire qualsiasi file dal router utilizzando l'FTP.

Verifica del risultato

In questa sezione vengono illustrati alcuni passaggi della sintassi del file.

1. La prima riga è la riga del `prefix`. Per l'esempio di `ifTable`, è:

```
prefix 1.3.6.1.2.1.2.2.1
```

Corrisponde all'OID dell'oggetto `ifEntry`. L'`ifTable` è composto da uno o più `ifEntries`.

2. La riga successiva elenca il numero di oggetti nella tabella. La riga è costituita dalla tabella delle parole chiave seguita dal numero di oggetti nella tabella, seguito dall'indice di ciascun oggetto. Ad esempio:

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

Questa riga indica che la tabella contiene 22 oggetti e che ogni oggetto ha un indice incrementale. Questi oggetti sono tratti dall'esempio `ifTable`:

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

3. Dopo questa riga sono presenti più voci di riga. Nell'esempio `ifTable`, ogni riga corrisponde a un'interfaccia. Le righe iniziano con la riga della parola chiave, seguita dal relativo identificatore di indice e dagli oggetti enumerati dalla voce di tabella precedente. Ad esempio:

```
row 1 1 546F6B656E52696E67302F30 9 4464 16000000 0008B0851800 2 2 6551 0 0 0 0 0 0 0 0 0  
0 0 0.0
```

4. La quarta voce è `ifDescr` per l'interfaccia 1. Tuttavia, si tratta di `ifDescr` in caratteri ASCII con codifica esadecimale. Per tradurre questa riga in un formato più leggibile, utilizzare questo comando Perl:

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

Questa voce corrisponde all'interfaccia `TokenRing0/0`. Tutti gli oggetti che normalmente sono stringhe vengono visualizzati come caratteri ASCII con codifica esadecimale nei file ausiliari. È possibile utilizzare questo comando Perl per convertire qualsiasi stringa ASCII esadecimale in testo leggibile. Se Perl non è disponibile, utilizzare questa [tabella di caratteri ASCII](#) per tradurre la stringa.

5. Alcune voci contengono caratteri `~` per i valori. Ciò significa che il valore dell'oggetto è NULL. In altre parole, l'oggetto non è istanziato sul dispositivo. Ad esempio:

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

Corrisponde all'interfaccia di livello `ATM1/0/0-atm`. Si noti che `ifMtu` è NULL per questa interfaccia. Poiché si tratta di un'interfaccia virtuale, ha senso che non abbia una MTU. Se si preferisce, è possibile sostituire questi valori NULL con `0` aggiungendo questo comando alla configurazione del dispositivo:

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

Risoluzione dei problemi relativi al risultato

Quando si esegue il polling dell'oggetto `cbfStatusFileState`, se si riceve un valore diverso da `running(1)`, `ready(2)` o `empted(3)`, si è verificato un errore. Le cause degli errori sono le seguenti:

```
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 il numero di oggetti nel file è inferiore al previsto, `cbfDefineMaxObjects` di CISCO-BULK-FILE-MIB potrebbe essere impostato su un valore troppo basso. Per determinare il valore corrente dell'oggetto, utilizzare **snmpget**.

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

Il valore 0 indica che non è configurato alcun limite. Il valore può essere impostato su qualsiasi numero intero compreso tra 0 e 4294967295 inclusi. Per impostare il numero massimo di oggetti per file su 10, utilizzare il comando **snmpset**. L'indice per questo oggetto è sempre 0.

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

Questo oggetto potrebbe non essere configurabile su tutte le piattaforme. Se l'operazione **snmpset** non riesce e viene restituito questo errore, l'oggetto non è configurabile sulla piattaforma in uso:

```
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
```

Durante il polling dell'oggetto `cfRequestResult`, se si riceve un valore diverso da `pending(1)` o `success(2)`, l'operazione FTP ha rilevato un errore. Le cause degli errori sono le seguenti:

```
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
```

Avvertenze

- Attualmente non è disponibile un metodo supportato per accedere direttamente ai file in blocco. Per leggere i file, occorre usare CISCO-FTP-CLIENT-MIB.
- L'oggetto `cbfDefineFileStorage` definisce tre tipi: `effimero`, `volatile` e `permanente`. Al momento, l'unico tipo supportato in IOS è `ephemeral`. I file temporanei esistono in piccole quantità fino a quando non vengono letti.
- Una volta letti, i file non possono essere riletti. Devono prima essere ricreati.
- L'oggetto `cbfDefineFileFormat` definisce tre tipi: `standardBER`, `bulkBinary` e `bulkASCII`. Gli unici formati supportati sono `bulkBinary` e `bulkASCII`. Il formato predefinito è `bulkBinary`.
- È noto che il server FTP Chameleon per Windows **non** funziona con CISCO-FTP-CLIENT-MIB, in quanto non restituisce codici di risultato corretti.

Informazioni correlate

- [Come configurare le stringhe della community SNMP su router, switch XL, RSM, MSFC e switch Catalyst basati su software Cisco IOS](#)
- [Supporto tecnico – Cisco Systems](#)