

# Comment utiliser CISCO-BULK-FILE-MIB

## Contenu

[Introduction](#)

[Avant de commencer](#)

[Conditions requises](#)

[Composants utilisés](#)

[Conventions](#)

[Informations générales](#)

[Utilisant le CISCO-BULK-FILE-MIB](#)

[Création d'une exécution BULK-FILE](#)

[Instructions pas à pas](#)

[Transférer le fichier utilisant le CISCO-FTP-CLIENT-MIB](#)

[Instructions pas à pas](#)

[Vérifier le résultat](#)

[Dépannage du résultat](#)

[Mises en garde](#)

[Informations connexes](#)

## Introduction

Ce document explique comment utiliser le CISCO-BULK-FILE-MIB et transférer des fichiers créés par ce Management Information Base (MIB) utilisant le CISCO-FTP-CLIENT-MIB.

À partir de la version de logiciel 12.0 de Cisco IOS®, Cisco a mis en application une manière d'enregistrer un objet de Protocole SNMP (Simple Network Management Protocol) ou de l'ajouter comme fichier sur le périphérique. Ce fichier peut alors être récupéré utilisant le CISCO-FTP-CLIENT-MIB. Cette technologie te permet pour transférer un grand nombre de données suivre une méthode de transport fiable.

## Avant de commencer

### Conditions requises

Avant de tenter cette configuration, assurez-vous que vous répondez à ces exigences :

- Vous avez une version de logiciel 12.0 ou ultérieures courante de Cisco IOS® de périphérique de Cisco. Vérifiez l'outil de localisateur MIB pour s'assurer que le CISCO-BULK-FILE-MIB est pris en charge pour votre périphérique. Un lien à l'outil peut être trouvé sur la [page Outils MIB de Cisco IOS](#). **Remarque:** Ce MIB n'est pas pris en charge sur des périphériques de SYSTÈME D'EXPLOITATION de Catalyst.

- Le SNMP doit être configuré sur le périphérique avec en lecture seule et des chaînes de caractères de la communauté en lecture-écriture. Ceci n'est pas couvert dans ce document. Pour les informations sur configurer le SNMP sur des périphériques IOS®, lu [comment configurer des chaînes de caractères de la communauté SNMP sur des Routeurs, des Commutateurs articulés autour d'un logiciel du Cisco IOS XL, RSMs, MSFCs et des Commutateurs de Catalyst.](#)

## Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Les CISCO-BULK-FILE-MIB pour enregistrer l'ifTable d'une exécution de 7507 routeurs 12.1(12) dans un fichier, emploient alors le CISCO-FTP-CLIENT-MIB pour transférer ce fichier du routeur vers un ftp server.
- La suite de commande SNMP net-[SNMP](#) installée sur l'UNIX ou le Windows.
- Ce MIB est utilisé :SNMPv2-TCSNMPv2-SMISNMPv2-CONFSNMPv2-MIBIANAIFTYPE-MIBIF-MIBCISCO-SMICISCO-TCCISCO-BULK-FILE-MIBCISCO-FTP-CLIENT-MIB

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

## Conventions

Pour plus d'informations sur les conventions de documents, reportez-vous à [Conventions relatives aux conseils techniques Cisco.](#)

## Informations générales

Veillez-vous pour avoir le MIB dans cette table chargée dans votre plate-forme d'administration. Ceci te permet pour utiliser les noms d'objet et les valeurs répertoriés ci-dessus au lieu des identifiants numériques d'objet (OID). Généralement ce document se rapporte à des noms d'objet et pas à des OID.

Format SMI de version 1	Format SMI de 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>

## Utilisant le CISCO-BULK-FILE-MIB

### Création d'une exécution BULK-FILE

Dans cet exemple, nous capturons l'`ifTable` d'un routeur, et l'enregistrons dans un fichier en vrac. Cependant, vous pouvez utiliser n'importe quel objet ou table MIB.

Utilisez la version net-SNMP du **snmpset**. L'adresse IP du routeur est **14.32.8.2**. Sa chaîne de caractères de la communauté en lecture-écriture est **privée**. La chaîne de caractères de la communauté en lecture seule est **publique**.

Chaque fois que vous créez une nouvelle exécution en vrac de fichier, choisissez deux nombres aléatoires pour l'exemple de ligne. Ils peuvent être tout nombre entre 1 et 4294967295 inclus. Aux fins de cet exemple, utilisez 333 et 444.

### Instructions pas à pas

Pour créer une exécution BULK-FILE, terminez-vous ces étapes :

1. Installez le fichier à créer.  

```
$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. Spécifiez l'objet MIB pour le capturer. Cet objet exige deux index pour l'exécution correcte. Les 333 est les 333 de la table de création de fichier ci-dessus. Les 444 est un nouveau nombre aléatoire utilisé pour l'index primaire dans le `cbfDefineObjectTable`. Cet exemple explique utilisant un nom d'objet pour le `cbfDefineObjectID` (`ifTable`). Vous pourriez utiliser un OID plein-qualifié ici aussi bien.

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

3. Lancez les lignes de création récente. Vous devez avoir les deux index pour votre ligne

```
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. Créez le fichier.  

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

Le fichier en vrac est créé.

5. Vérifiez le fichier a été créé avec succès à l'aide du **snmpget** sur l'objet de `cbfStatusFileState`. Cet objet exige deux index. Le premier index est le nombre aléatoire choisi pour la table de fichier (333 dans cet exemple). Le deuxième index dépend de combien de fichiers vous avez créés dans votre routeur. Puisque c'est votre premier fichier, l'index est 1. , Utilisez par conséquent la commande :

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

Une valeur de `running(1)` signifie que le fichier est en cours d'être créée. Une valeur de `ready(2)` signifie que le fichier a été créé avec succès, et attend d'être lue.

Ce fichier n'est pas directement accessible du routeur, cependant. Employez le CISCO-FTP-CLIENT-MIB pour lire ce fichier.

### Transférer le fichier utilisant le CISCO-FTP-CLIENT-MIB

Pour chaque exécution de client FTP, vous devez sélectionner un nombre aléatoire pour l'exemple de ligne. Vous pouvez utiliser un des mêmes nombres aléatoires que vous avez utilisés en haut. Cet exemple utilise 555.

## Instructions pas à pas

Pour transférer le fichier utilisant un CISCO-FTP-CLIENT-MIB, terminez-vous ces étapes :

1. Créez un exemple de ligne du client FTP.  

```
$ snmpset -c private 14.32.8.2 cfcRequestEntryStatus.555 i 5
```
2. Complétez les paramètres requis. Le LocalFile **doit** être le même nom que le fichier que vous avez créé en haut ! **PutASCII** d'utilisation pour transférer des fichiers de `bulkASCII`. Si vous placez le `cbfDefineFileFormat` à ci-dessus `bulkBinary`, vous devez placer le `cfcRequestOperation` à `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. Commencez le transfert en plaçant la ligne à l'active.  

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

 Le transfert de FTP commence. Si complet, le fichier est enregistré à `/home/Marcus/ifTable.txt`.
4. Pour obtenir le statut du transfert de FTP, **snmpget** d'utilisation de nouveau sur l'objet de `cfcRequestResult`. Cet objet utilise le même index que vous avez utilisé avec les autres objets de FTP.  

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

 Une valeur de `pending(1)` signifie que le fichier transfère toujours. Une valeur de `success(2)` signifie le fichier transféré avec succès. N'importe quelle autre valeur est une [erreur](#).
5. Quand le fichier est transféré fait, essayez le **snmpget** de l'objet de `cbfStatusFileState` de nouveau. Il a maintenant une valeur différente.  

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

```
enterprises.cisco.ciscoMgmt.ciscoBulkFileMIB.ciscoBulkFileMIBObjects.cbfStatus.cbfStatusFileTable.cbfStatusFileEntry.cbfStatusFileState.333.1 = emptied(3)
```

 La valeur d'`emptied(3)` signifie que le fichier a été avec succès indiqué. Le fichier ne peut pas être transféré de nouveau.
6. Il est maintenant sûr de supprimer ce fichier en détruisant la ligne d'état de fichier. Cet objet prend les mêmes index que le `cbfStatusFileState` ci-dessus.  

```
$ snmpset -c private 14.32.8.2 cbfStatusFileEntryStatus.333.1 i 6
```
7. Une fois le fichier est supprimé, supprime l'objet correspondant et classe des lignes.  

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

De cette manière, vous pouvez employer le CISCO-FTP-CLIENT-MIB pour transférer n'importe quel fichier hors fonction du routeur utilisant le FTP.

## Vérifier le résultat

Cette section vous guide en lisant une partie de la syntaxe pour ce fichier.

1. La première ligne est la ligne de `préfixe`. Pour notre exemple `ifTable`, il est :  

```
prefix 1.3.6.1.2.1.2.2.1
```

 Ceci correspond à l'OID pour l'objet `ifEntry`. L'`ifTable` se compose d'un ou plusieurs `ifEntries`.

2. La prochaine ligne répertorie le nombre d'objets dans la table. La ligne comprend la table de mot clé suivie du nombre d'objets dans la table, suivi de l'index de chaque objet. Exemple :
 

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

 Ce les états de la ligne que la table contient 22 objets, et chaque objet a un index de incrémentation. Ces objets sont de l'exemple `ifTable` :
 

```
ifIndex
ifDescr
ifType
ifSpeed
...
```
3. Après cette ligne, il y a de plusieurs entrées de ligne. Dans l'exemple `ifTable`, chaque ligne correspond à une interface. Les lignes commencent par la ligne de mot clé, suivie de leur identifiant d'index, et suivie des objets énumérés par l'entrée de table précédente. Exemple :
 

```
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 quatrième entrée est l'`ifDescr` pour l'interface 1. Cependant, c'est l'`ifDescr` dans l'ASCII encodée par hexadécimal. Pour traduire cette ligne en format plus accessible en lecture, utilisez cette commande Perl :
 

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

 TokenRing0/0 Cette entrée correspond pour relier TokenRing0/0. Tous les objets qui sont normalement des chaînes sont affichés en tant qu'ASCII encodée par hexadécimal dans les fichiers en vrac. Vous pouvez utiliser cette commande Perl de traduire n'importe quelle chaîne hexadécimale ASCII en texte accessible en lecture. Si vous n'avez pas le Perl, employez cette [table de caractère ASCII](#) pour traduire la chaîne.
5. Exposition de quelques entrées | caractères pour des valeurs. Ceci signifie que la valeur pour cet objet est NULLE. C'est-à-dire, l'objet pas instancié sur le périphérique. Exemple :
 

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

 Ceci correspond à l'interface de couche ATM1/0/0-atm. Notez que l'`ifMtu` est NUL pour cette interface. Puisque c'est une interface virtuelle, il semble raisonnable qu'il n'a pas un MTU. Si vous préférez, vous pouvez remplacer ces nulls par 0 en ajoutant cette commande à la configuration de périphérique :
 

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

## Dépannage du résultat

En votant l'objet de `cbfStatusFileState`, si vous recevez une valeur autre que `running(1)`, `ready(2)`, ou `emptied(3)`, votre exécution a rencontré une erreur. Ce sont des causes pour les erreurs :

```
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 le nombre d'objets dans le fichier est moins que vous prévoyez, des `cbfDefineMaxObjects` du CISCO-BULK-FILE-MIB peuvent être placés si bas. Pour déterminer la valeur courante de l'objet, `snmpget` d'utilisation.

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

Une valeur de 0 signifie qu'aucune limite n'est configurée. La valeur peut être placée à n'importe quel entier entre 0 et 4294967295, inclus. Pour placer les objets de maximum par fichier à 10, utilisez la commande de `snmpset`. L'index pour cet objet est toujours 0.

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

Cet objet peut ne pas être configurable sur toutes les Plateformes. Si le **snmpset** échoue avec cette erreur, l'objet n'est pas configurable sur votre plate-forme :

```
Error in packet.
```

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

```
Failed object:
```

```
enterprises.cisco.ciscoMgmt.ciscoBulkFileMIB.ciscoBulkFileMIBObjects.cbfDefine.cbfDefineMaxObjects.0
```

En votant l'objet de `cfcRequestResult`, si vous recevez une valeur autre que `pending(1)` ou `success(2)`, l'exécution de FTP a rencontré une erreur. Ce sont des causes pour les erreurs :

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

## Mises en garde

- Actuellement il n'y a aucune manière prise en charge d'accéder aux fichiers en vrac directement. Vous devez passer par le CISCO-FTP-CLIENT-MIB pour lire les fichiers.
- L'objet de `cbfDefineFileStorage` définit trois types : éphémère, volatil, et constante. Actuellement, le seul type pris en charge dans l'IOS est éphémère. Les fichiers éphémères existent dans un peu jusqu'à indiqué.
- Une fois que les fichiers sont indiqués, ils ne peuvent pas être relus. Ils doivent d'abord être recréés.
- L'objet de `cbfDefineFileFormat` définit trois types : `standardBER`, `bulkBinary`, et `bulkASCII`. Les seuls formats pris en charge sont `bulkBinary` et `bulkASCII`. Le format par défaut est `bulkBinary`.
- Le ftp server de caméléon pour Windows est connu pour ne pas fonctionner avec le CISCO-FTP-CLIENT-MIB, puisqu'il ne renvoie pas des codes corrects de résultat.

## Informations connexes

- [Comment configurer des chaînes de caractères de la communauté SNMP sur des Routeurs, des Commutateurs articulés autour d'un logiciel du Cisco IOS XL, RSMs, MSFCs et des Commutateurs de Catalyst](#)
- [Support technique - Cisco Systems](#)