Automatisation des séquences de commandes Cisco IOS XE à l'aide de TCL et EEM

Table des matières

- Introduction
- Conditions préalables
  - Exigences
  - Composants utilisés
- Informations générales
- Configuration et procédure pas à pas
  - Étape 1. Création d'un fichier texte avec une séquence de commandes
    - Création d'un fichier texte à l'aide de scripts TCL
  - Étape 2. Exécution du fichier à l'aide du script EEM
    - Création du script EEM
    - Explication du script EEM
    - Exécution du script EEM
    - Résultat de l'exécution du fichier
- Résumé

Introduction

Ce document décrit une méthode pour créer un script EEM unique et unique capable d'exécuter n'importe quelle séquence de commandes telle que définie par un administrateur réseau.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Connaissance des opérations de Cisco IOS® XE
- Présentation des scripts EEM (Embedded Event Manager)
- Connaissance des commutateurs Cisco Catalyst
- Connaissance de base des scripts TCL

Composants utilisés

Les informations contenues dans ce document sont basées sur le commutateur Cisco Catalyst 9600X (CS-C9600X) exécutant la version IOS XE 17.18.2.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l’incidence possible des commandes.

Informations générales

L'administrateur crée un fichier texte brut contenant la séquence de commandes souhaitée. Le script EEM appelle ensuite ce fichier sur un déclencheur spécifié et exécute toutes les commandes répertoriées dans le fichier texte.

Configuration et procédure pas à pas

Étape 1. Création d’un fichier texte avec une séquence de commandes

1. Créez un fichier texte contenant les commandes requises dans l'ordre souhaité.
2. Copiez le fichier dans la mémoire flash du périphérique.
3. Utilisez les scripts FTP, TFTP ou TCL pour créer ou transférer le fichier.

Création d'un fichier texte à l'aide de scripts TCL

Syntaxe:

<#root>

```
tclsh
puts [open "flash:COMMANDS_TO_EXECUTE.txt" w+] {


Enter Commands to Execute



}
exit
```

Exemple de configuration :

Afin de créer une séquence de commandes qui arrête l'interface FiftyGigE1/1/0/3 :

```
C9600-Silicon1-SVL#tclsh
C9600-Silicon1-SVL(tcl)#puts [open "flash:COMMANDS_TO_EXECUTE.txt" w+] {
configure terminal
interface fiftyGigE 1/1/0/3
shutdown
end
show run interface fiftyGigE 1/1/0/3
}
C9600-Silicon1-SVL(tcl)#exit
C9600-Silicon1-SVL#
```

Vérifiez le fichier dans la mémoire flash :

```
C9600-Silicon1-SVL#dir flash:
Directory of flash:/

357117 -rw- 101 Jan 25 2026 21:37:09 +05:30 COMMANDS_TO_EXECUTE.txt
```

Contenu de COMMANDS_TO_EXECUTE.txt

```
configure terminal
interface fiftyGigE 1/1/0/3
shutdown
end
show run interface fiftyGigE 1/1/0/3
```

Ce fichier peut être créé par script TCL ou transféré à l'aide de FTP, TFTP ou RCP. Le fichier doit être modifié selon les besoins afin d'inclure toute séquence de commandes requise par l'administrateur réseau.

Étape 2. Exécution du fichier à l’aide du script EEM

Création du script EEM

Cette applet EEM exécute toutes les commandes du fichier spécifié COMMANDS_TO_EXECUTE.

```
event manager applet Cisco_Live authorization bypass
event none maxrun 1000
```

```
action 010 cli command "enable"
action 020 cli command "term exec prompt timestamp"
action 030 cli command "term exec prompt expand"
action 040 file open fileout1 flash:OUTPUT.txt a+
action 050 file open filein1 flash:COMMANDS_TO_EXECUTE.txt r
action 060 file read filein1 input
action 070 foreach line $input "\n"
action 080 cli command "$line"
action 090 file write fileout1 "$_cli_result"
action 100 end
action 110 file close fileout1
action 120 file close filein1
```

Explication du script EEM

++ applet du gestionnaire d'événements Contournement d'autorisation Cisco_Live :
Nom applet applet du gestionnaire d'événements : Enregistre une applet EEM portant le nom spécifié (« Cisco_Live » dans ce cas).
Contournement d'autorisation : Les mots-clés facultatifs qui spécifient l'applet doivent contourner l'autorisation AAA.

++ événement aucun maxrun 1000 :
Événement aucun : Cela indique que la stratégie EEM est enregistrée avec le détecteur d'événements « none », ce qui signifie qu'elle n'est pas déclenchée par un événement automatique, mais qu'elle doit être exécutée manuellement.
Maxrun 1000 : Ce paramètre facultatif définit la durée d'exécution maximale de l'applet ou de la stratégie sur 1 000 secondes. S'il n'est pas spécifié, le délai d'exécution maximal par défaut est de 20 secondes.

++ action 020 cli command term exec prompt timestamp :
Action 200 : Il s'agit du libellé ou de l'identificateur de séquence de l'action dans le script EEM.
term exec prompt timestamp est une commande CLI qui active les horodatages sur l’invite dans la session de terminal en cours.
Cette fonctionnalité facilite le dépannage et l'audit en fournissant une référence temporelle pour chaque commande exécutée.
L'horodatage apparaît avant la sortie des commandes, ce qui facilite la corrélation des événements et des exécutions de commandes avec des heures spécifiques.

++ action 030 cli command term exec prompt développement :
term exec prompt expand active le mode d’invite développé pour la session de terminal en cours.
Et il configure le terminal pour afficher l'invite complète sans troncature ni abréviation.
Il est utile dans les scripts ou les scénarios de dépannage où le contexte d'invite complet est

nécessaire pour des raisons de clarté ou d'analyse.
La commande affecte uniquement la session de terminal en cours et ne persiste pas après la fin de la session.

++ action 040 file open fileout1 flash:OUTPUT.txt a+:
Fichier ouvert : Indique l'opération d'ouverture d'un fichier.
Fileout1 : Il s'agit du descripteur de fichier ou du nom de handle utilisé dans le script pour faire référence au fichier ouvert.
Flash : OUTPUT.txt : Il s'agit du chemin d'accès et du nom du fichier dans le système de fichiers flash où le fichier réside ou sera créé.
a+ : Il s'agit du mode d'accès au fichier, ce qui signifie que le fichier est ouvert pour lecture et ajout. Si le fichier n'existe pas, il sera créé. Les données écrites dans le fichier seront ajoutées à la fin sans tronquer le contenu existant.

++ action 050 file open file in1 flash:COMMANDS_TO_EXECUTE.txt r:
Fichier1 : Il s'agit du descripteur de fichier ou du nom de handle utilisé dans le script pour faire référence au fichier ouvert.
Flash : COMMANDS_TO_EXECUTE.txt : Il s'agit du chemin d'accès et du nom du fichier dans le système de fichiers flash où réside le fichier.
r : Il s'agit du mode d'accès au fichier, ce qui signifie que le fichier est ouvert en lecture seule.

++ action 060 file read file in1 input :
Lecture du fichier : Indique l'opération à effectuer pour lire un fichier.
Fichier1 : Il s'agit du descripteur ou de l'identificateur de fichier qui a été attribué lors de l'ouverture du fichier (par exemple, avec l'action 050 file open file in1. Flash : COMMANDS_TO_EXECUTE.txt r).
Entrée: Il s'agit du nom de la variable dans laquelle les données lues seront stockées pour être utilisées dans les actions de script suivantes.

++ action 070 pour chaque ligne $input '\n':
Foreach : Cette commande spécifie une itération sur une chaîne.
Ligne : Il s'agit de la variable d'itérateur qui contiendra chaque jeton (ligne) pendant chaque itération.
$input : Il s'agit de la variable de chaîne d'entrée contenant plusieurs lignes ou entrées.
'\n': Délimiteur utilisé pour diviser la chaîne d'entrée en jetons, dans ce cas, le caractère de nouvelle ligne.
Chaque ligne est un jeton et \n est le séparateur dans ce cas.

++ action 080 cli command $line :
Commande Cli : Indique que l'action consiste à exécuter une commande CLI sur le périphérique.
$line : Variable contenant la chaîne de commande CLI à exécuter. Les guillemets garantissent que la chaîne entière est traitée comme une seule commande.

++ action 090 file write fileout1 '$_cli_result' :
Écriture du fichier : La commande EEM pour écrire des données dans un fichier.
Fileout1 : Le gestionnaire de fichiers précédemment ouvert par une commande telle que file open fileout1 flash:OUTPUT.txt a+, représentant le fichier dans lequel les données sont écrites.
'$_cli_result' : Variable contenant la sortie de la commande CLI ou toute donnée de chaîne à écrire dans le fichier. Les guillemets garantissent que l'intégralité du contenu est traitée comme une seule chaîne.
Cette commande est généralement utilisée après l'exécution des commandes CLI et la capture de leur résultat dans $_cli_result, ce qui permet au script de consigner ou d'archiver le résultat dans un fichier sur le stockage flash du périphérique.

++ action 100 fin :
Fin : Mot clé qui signifie la fin d'un bloc conditionnel tel que if, else ou while.

++ action 120 fichier fermer le fichier1 :
Fermeture du fichier : Opération de fermeture du descripteur de fichier.
Fichier1 : Identificateur du descripteur de fichier à fermer, qui doit avoir été ouvert plus tôt dans le script.

Exécution du script EEM

Afin de déclencher manuellement le script EEM :

```
C9600-Silicon1-SVL#event manager run Cisco_Live
```

Dans cet exemple, le script EEM est déclenché manuellement. Le script peut avoir n'importe quel déclencheur selon les exigences du réseau.

Résultat de l'exécution du fichier

L'affichage du fichier de sortie affiche les commandes exécutées et leurs résultats :

<#root>

```
C9600-Silicon1-SVL#more flash:OUTPUT.txt

Enter configuration commands, one per line. End with CNTL/Z.
C9600-Silicon1-SVL(config)#
C9600-Silicon1-SVL(config-if)#
C9600-Silicon1-SVL(config-if)#
C9600-Silicon1-SVL#Load for five secs: 0%/0%; one minute: 1%; five minutes: 1%
No time source, *22:01:35.406 IST Sun Jan 25 2026
------------------ show running-config interface FiftyGigE1/1/0/3 ------------------

Building configuration...

Current configuration : 127 bytes
!
interface FiftyGigE1/1/0/3
description To-Host3-Ten 1/1/3
switchport access vlan 10
switchport mode access

shutdown


end
```

L’interface FiftyGigE1/1/0/3 est arrêtée avec succès selon la séquence de commandes.

Résumé

Ce document a montré comment combiner les scripts TCL et l'EEM de Cisco afin de créer un script EEM unique et flexible capable d'exécuter n'importe quelle séquence de commandes stockées dans un fichier texte. Cette approche permet aux administrateurs réseau de définir des séquences de commandes en externe et de déclencher leur exécution automatiquement ou manuellement, améliorant ainsi l'automatisation et l'efficacité opérationnelle sur les commutateurs Cisco Catalyst exécutant Cisco IOS XE.

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.