

# Automazione delle sequenze di comandi Cisco IOS XE tramite TCL ed EEM

## Introduzione

In questo documento viene descritto un metodo per creare uno script EEM univoco e singolo in grado di eseguire qualsiasi sequenza di comandi definita da un amministratore di rete.

## Prerequisiti

### Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Conoscenza delle operazioni di Cisco IOS® XE
- Informazioni sugli script di Embedded Event Manager (EEM)
- Familiarità con gli switch Cisco Catalyst
- Conoscenze base di scripting TCL

## Componenti usati

Per la stesura del documento, sono stati usati switch Cisco Catalyst serie 9600X (CS-C9600X) con software IOS XE 17.18.2.

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.

## Premesse

L'amministratore crea un file di testo normale contenente la sequenza di comandi desiderata. Lo script EEM chiama quindi questo file su un trigger specificato ed esegue tutti i comandi elencati nel file di testo.

# Configurazione e procedura dettagliata

## Passaggio 1. Creazione di un file di testo con una sequenza di comandi

1. Create un file di testo contenente i comandi richiesti nella sequenza desiderata.
2. Copiare il file nella memoria flash del dispositivo.
3. Usate gli script FTP, TFTP o TCL per creare o trasferire il file.

### Creazione di un file di testo mediante script TCL

#### Sintassi:

```
<#root>
```

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

```
Enter Commands to Execute
```

```
}  
exit
```

#### Esempio di configurazione:

Per creare una sequenza di comandi che chiudano l'interfaccia 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#
```

Verificare il file nella memoria flash:

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

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

## Contenuto di COMMANDS\_TO\_EXECUTE.txt

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

Questo file può essere creato tramite script TCL o trasferito tramite FTP, TFTP o RCP. Il file deve essere modificato in base alle esigenze per includere qualsiasi sequenza di comandi richiesta dall'amministratore di rete.

## Passaggio 2. Eseguire il file utilizzando lo script EEM

### Creazione dello script EEM

Questa applet EEM esegue tutti i comandi dal file specificato 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
```

### Spiegazione dello script EEM

++ applet per la gestione degli eventi Cisco\_Live bypass autorizzazioni:

Nome applet gestione eventi: Registra un'applet EEM con il nome specificato ('Cisco\_Live' in

questo caso).

Bypass autorizzazione: Le parole chiave opzionali che specificano l'applet devono ignorare l'autorizzazione AAA.

Evento + none maxrun 1000:

Evento nessuno: In questo modo si specifica che il criterio EEM è registrato con il rilevatore eventi 'none', ovvero non viene attivato da alcun evento automatico ma deve essere eseguito manualmente.

Maxrun 1000: Questo parametro facoltativo imposta su 1000 secondi il tempo di esecuzione massimo per l'applet o il criterio. Se non viene specificato, il runtime massimo predefinito è 20 secondi.

++ azione 020 cli termine exec prompt timestamp:

Azione 020: Etichetta o identificatore di sequenza per l'azione nello script EEM.

term exec prompt timestamp è un comando CLI che abilita timestamp sul prompt nella sessione corrente del terminale.

Questa funzione facilita la risoluzione dei problemi e il controllo fornendo un riferimento temporale per ciascun comando eseguito.

L'indicatore orario viene visualizzato prima dell'output dei comandi, semplificando la correlazione di eventi ed esecuzioni di comandi con orari specifici.

++ action 030 cli termine exec prompt expand:

term exec prompt expand attiva la modalità di prompt espansa per la sessione terminale corrente e configura il terminale in modo da visualizzare il prompt completo senza troncamento o abbreviazione.

Aiuta negli script o nella risoluzione di scenari in cui è necessario il contesto del prompt completo per la chiarezza o l'analisi.

Il comando influisce solo sulla sessione terminale corrente e non viene mantenuto al termine della sessione.

++ azione 040 file apri fileout1 flash:OUTPUT.txt a+:

Apertura file: Specifica l'operazione di apertura di un file.

Fileout1: Descrittore di file o nome di handle utilizzato nello script per fare riferimento al file aperto.

Flash:OUTPUT.txt: Si tratta del percorso e del nome del file nel file system flash in cui il file risiede o verrà creato.

a+: Questa è la modalità di accesso al file, che indica che il file è aperto per la lettura e l'aggiunta. Se il file non esiste, verrà creato. I dati scritti nel file verranno aggiunti alla fine senza troncamento il contenuto esistente.

++ azione 050 file apri filein1 flash:COMMANDS\_TO\_EXECUTE.txt r:

Filein1: Descrittore di file o nome di handle utilizzato nello script per fare riferimento al file aperto.

Flash:COMMANDS\_TO\_EXECUTE.txt: Si tratta del percorso e del nome del file nel file system flash in cui risiede il file.

r: Questa è la modalità di accesso al file, che indica che il file è aperto in sola lettura.

++ azione 060 file letta file in input1:

Lettura file: Specifica l'operazione di lettura da un file.

Filein1: Si tratta del descrittore o dell'handle di file assegnato quando il file è stato aperto, ad esempio con l'azione 050 file aperto in1. Flash:COMMANDS\_TO\_EXECUTE.txt r.

Ingresso: Nome della variabile in cui verranno archiviati i dati letti per l'utilizzo in azioni di script successive.

++ azione 070 per ogni riga \$input '\n':

Foreach: Questo comando specifica un'iterazione su una stringa.

Riga: Variabile iteratore che conterrà ogni token (riga) durante ogni iterazione.

\$input: Variabile stringa di input contenente più righe o voci.

'\n': Delimitatore utilizzato per dividere la stringa di input in token, in questo caso il carattere di nuova riga.

Ogni riga è un token e \n in questo caso è la barra di divisione.

++ action 080 cli comando \$line:

Comando Cli: Specifica che l'azione consiste nell'eseguire un comando CLI sul dispositivo.

\$riga: Variabile contenente la stringa di comando CLI da eseguire. Le virgolette garantiscono che l'intera stringa venga considerata come un unico comando.

++ azione 090 file write fileout1 '\$\_cli\_result':

Scrittura file: Comando EEM per scrivere dati in un file.

Fileout1: Gestore di file precedentemente aperto da un comando come file open fileout1

flash:OUTPUT.txt a+, che rappresenta il file in cui vengono scritti i dati.

'\$\_cli\_result': La variabile che contiene l'output del comando CLI o qualsiasi dato stringa da scrivere nel file. Le virgolette garantiscono che l'intero contenuto venga considerato come una singola stringa.

Questo comando viene in genere utilizzato dopo l'esecuzione dei comandi CLI e l'acquisizione dell'output in \$\_cli\_result, consentendo allo script di registrare o archiviare l'output in un file nella memoria flash del dispositivo.

Fine azione ++ 100:

End: Parola chiave che indica la fine di un blocco condizionale, ad esempio if, else o while.

++ azione 120 file chiudifilein1:

Chiusura file: Operazione di chiusura del descrittore di file.

Filein1: Identificatore del descrittore di file da chiudere, che deve essere stato aperto in precedenza nello script.

Esecuzione script EEM

Per attivare manualmente lo script EEM:

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

In questo esempio lo script EEM viene attivato manualmente. Lo script può avere qualsiasi trigger in base ai requisiti di rete.

Risultato dell'esecuzione dei file

La visualizzazione del file di output mostra i comandi eseguiti e i relativi risultati:

```
<#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'interfaccia FiftyGigE1/1/0/3 è stata chiusa correttamente dalla sequenza di comandi.

## Riepilogo

In questo documento viene illustrato come combinare gli script TCL con l'EEM di Cisco per creare uno script EEM singolo e flessibile in grado di eseguire qualsiasi sequenza di comandi archiviata in un file di testo. Questo approccio consente agli amministratori di rete di definire le sequenze di comandi esternamente e di attivarne l'esecuzione automaticamente o manualmente, migliorando l'automazione e l'efficienza operativa sugli switch Cisco Catalyst con Cisco IOS XE.

## Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).