

# Esempio di configurazione del modulo CLI della shell Cisco IOS

## Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Configurazione](#)

[Tubi e reindirizzamento](#)

[Applet integrate](#)

[Condizioni](#)

[Loop](#)

[Operazioni sulle stringhe](#)

[Esempio di configurazione](#)

[Verifica](#)

[Risoluzione dei problemi](#)

[Informazioni correlate](#)

## Introduzione

In questo documento viene descritto il modulo CLI di Cisco IOS<sup>®</sup> Shell (IOS.sh).

## Prerequisiti

### Requisiti

Cisco raccomanda la conoscenza di Cisco IOS versione 15.1(4)M, 15.1(2)S e successive. Per utilizzare le funzionalità e le funzioni di Cisco IOS.sh sul router, è necessario configurare e abilitare Cisco IOS.sh.

### Componenti usati

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

- Switch Cisco Catalyst serie 6500

- Cisco IOS release 15.1(1)SY

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

Cisco IOS.sh è un modulo che consente di utilizzare gli script della shell nella CLI di Cisco IOS. Questo modulo consente di automatizzare le azioni di routine e semplifica la gestione dei dispositivi. Può essere utilizzato in combinazione con altri strumenti di gestione, ad esempio Cisco IOS Embedded Event Manager (EEM).

## Configurazione

**Nota:** per ulteriori informazioni sui comandi menzionati in questa sezione, usare lo [strumento di ricerca dei comandi \(solo utenti registrati\)](#).

Per abilitare Shell CLI sul dispositivo, immettere:

```
Switch#terminal shell
```

Per disabilitare Shell CLI, immettere:

```
Switch#terminal no shell
```

Questa funzionalità supporta la variabile PATH, che contiene un elenco di directory in cui la shell può cercare funzioni/applet. Inoltre, Shell CLI supporta le seguenti funzionalità:

## Tubi e reindirizzamento

```
Switch#show version | grep 15.1
Cisco IOS Software, s2t54 Software (s2t54-ADVENTERPRISEK9-M),
Version 15.1(1)SY, RELEASE SOFTWARE (fc2)
```

```
Switch#show version | grep 15 > bootdisk:version.txt
Switch#dir bootdisk:
Directory of bootdisk:/
```

```
14 -rw- 280 Nov 15 2012 23:25:32 +00:00 version.txt
```

```
1024557056 bytes total (577273856 bytes free)
```

```
Switch#cat bootdisk:version.txt
Cisco IOS Software, s2t54 Software (s2t54-ADVENTERPRISEK9-M),
Version 15.1(1)SY, RELEASE SOFTWARE (fc2)
```

## Applet integrate

Cisco IOS.sh contiene diverse applet incorporate che consentono di creare applicazioni e filtri per le funzioni utente:

```
[[           evaluate a logical test expression
cat         output data from a pipe or file to the terminal
cut         edit piped output
echo        echo arguments to the terminal
false      return false in while or if expressions, and set the result
fetch      return values from the configuration database
grep       search for regular expressions in piped output or files
head       print the first lines in the input
interface  print interfaces that match the argument
let        evaluate a numeric expression, and set the result
man        print information for built-ins
more       page piped output to the terminal
nl         number the lines in the input
null       ignore the input
printf     output formatted data to the terminal
read       read input into variables
set_oper   set operational values
sleep      pause execution of the terminal
sort       sort the input
tail       print the tail of the input
true      return true in while or if expressions, and set the result
uname     print system information
wc        count lines, words, and characters
```

## Condizioni

Di seguito sono riportati i comandi per il **se...Allora....else**.

```
Switch#x=5
Switch#if [[ $x != 5 ]];
then.else.fi>then
then.else.fi>echo false;
then.else.fi>else
else..fi>echo true;
else..fi>fi;
true
Switch#
```

Questi operatori di confronto vengono utilizzati per utilizzare valori interi:

### Operatori Descrizione

-eq	Arg1-eq Arg2. True se Arg1 è uguale a Arg2
-ne	Arg1-ne Arg2. True se Arg1 è diverso da Arg2
-lt	Arg1-lt Arg2. True se Arg1 è minore di Arg2
-gt	Arg1-gt Arg2. True se Arg1 è maggiore di Arg2
-ge	Arg1-ge Arg2. True se Arg1 è maggiore o uguale a Arg2
-le	Arg1-le Arg2. True se Arg1 è minore o uguale a Arg2

Per utilizzare i file, sono disponibili le seguenti condizioni:

Operatori	Descrizione
-a oppure -e	True se il file esiste
-d	True se il file esiste ed è una directory

-f	True se il file esiste ed è un file normale
-r	True se il file esiste ed è leggibile
-s	True se il file esiste e ha dimensioni maggiori di zero
-w	True se il file esiste ed è eseguibile
-nt	Verifica se file1 è più recente di file2. Per questo confronto viene utilizzata la data di modifica file
-ot	Verifica se file1 è precedente a file2

Per utilizzare diverse condizioni all'interno di un operatore di confronto, è possibile utilizzare i seguenti operatori logici:

```
&&    Logical AND
||     Logical OR
```

## Loop

I loop sono gli stessi della shell Linux. Sono disponibili due operatori.

Di seguito sono riportati i comandi per l'operatore **for**:

```
Switch#for x in 11
do..done>do
do..done>ping 192.0.2.$x
do..done>done
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.0.2.10, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Switch#
```

Di seguito sono riportati i comandi per l'operatore **while**:

```
Switch#for i in 1 2 3
do..done>do
do..done>while [ $i -lt 3 ]
do..done>do
do..done>echo $i
do..done>done
do..done>i=i+1
do..done>done
Switch#
```

## Operazioni sulle stringhe

Per i confronti tra stringhe, il comando **if...Allora....else** può essere utilizzato con le seguenti modifiche:

Operatori Descrizione

==	String 1 == string2, True se le stringhe sono uguali.
!=	String1 != string2 , True se le stringhe non sono uguali
<	String1 < string2, True se string1 è lessicograficamente minore di string2
>	Stringa1 > stringa2, True se stringa1 è lessicograficamente maggiore di stringa2

# Esempio di configurazione

```
Switch#function add_desc() {  
{..} >show running-config interface Gi$1  
{..} >configuration terminal  
{..} >interface Gi$1  
{..} >description $2  
{..} >end  
{..} >show running-config interface Gi$1  
{..} >}  
Switch#add_desc 5/1 SHELL_TEST  
Building configuration...
```

Current configuration : 74 bytes

```
!  
interface GigabitEthernet5/1  
ip address 192.0.2.10 255.255.255.0  
end
```

Enter configuration commands, one per line. End with CNTL/Z.  
Building configuration...

Current configuration : 98 bytes

```
!  
interface GigabitEthernet5/1  
description SHELL_TEST  
ip address 192.0.2.10 255.255.255.0  
end
```

Switch#

## Verifica

Fare riferimento a questa sezione per verificare che la configurazione funzioni correttamente.

Immettere il comando **show terminal** per visualizzare la configurazione corrente del terminale:

```
Switch#show terminal  
Line 2, Location: "", Type: "XTERM-COLOR"  
Length: 44 lines, Width: 155 columns  
Baud rate (TX/RX) is 9600/9600  
Status: PSI Enabled, Ready, Active, No Exit Banner, Ctrl-c Enabled  
.....  
Preferred transport is lat.  
Shell: enabled  
Shell trace: off
```

Immettere il comando **show shell functions** per visualizzare tutte le funzioni disponibili nel modulo, incluse le funzioni definite dall'utente:

```
Switch#show shell functions  
#User defined functions:  
  
Function namespace: DEFAULT  
function add_desc()  
{
```

```
show running-config interface Gi$1
configure terminal
interface Gi$1
description $2
end
show running-config interface Gi$1
}
```

## Risoluzione dei problemi

Al momento non sono disponibili informazioni specifiche per la risoluzione dei problemi di questa configurazione.

## Informazioni correlate

- [Guide alla configurazione di Catalyst 6500](#)
- [Documentazione e supporto tecnico – Cisco Systems](#)