

Cisco IOS Shell CLI 모듈 컨피그레이션 예

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[구성](#)

[파이프 및 리디렉션](#)

[임베디드 애플릿](#)

[조건](#)

[루프](#)

[문자열 작업](#)

[컨피그레이션 예시](#)

[다음을 확인합니다.](#)

[문제 해결](#)

[관련 정보](#)

소개

이 문서에서는 Cisco IOS[®] Shell(IOS.sh) CLI 모듈에 대해 설명합니다.

사전 요구 사항

요구 사항

Cisco에서는 Cisco IOS Release 15.1(4)M, 15.1(2)S 및 이후 릴리스에 대한 지식을 보유하고 있는 것이 좋습니다. 라우터에서 Cisco IOS.sh 기능 및 기능을 사용하려면 Cisco IOS.sh를 구성하고 활성화해야 합니다.

사용되는 구성 요소

이 문서의 정보는 다음 하드웨어 및 소프트웨어 버전을 기반으로 합니다.

- Cisco Catalyst 6500 Series 스위치
- Cisco IOS 릴리스 15.1(1)SY

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스

이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우, 모든 명령어의 잠재적인 영향을 미리 숙지하시기 바랍니다.

배경 정보

Cisco IOS.sh는 Cisco IOS CLI에서 셸 스크립팅을 사용할 수 있는 모듈입니다. 이 모듈은 일상적인 작업을 자동화하고 장치 관리를 간소화합니다. Cisco IOS Embedded Event Manager(EEM)와 같은 다른 관리 툴과 함께 사용할 수 있습니다.

구성

참고: 이 [섹션](#)에 사용된 명령에 대한 자세한 내용을 보려면 [Command Lookup Tool](#)([등록된 고객만 해당](#))을 사용합니다.

디바이스에서 셸 CLI를 활성화하려면 다음을 입력합니다.

```
Switch#terminal shell
```

셸 CLI를 비활성화하려면 다음을 입력합니다.

```
Switch#terminal no shell
```

이 기능은 셸이 함수/애플릿을 검색할 수 있는 디렉토리 목록을 포함하는 PATH 변수를 지원합니다. 또한 셸 CLI는 다음 기능을 지원합니다.

파이프 및 리디렉션

```
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)
```

임베디드 애플릿

Cisco IOS.sh에는 사용자 기능을 위한 애플리케이션 및 필터를 생성하는 데 도움이 되는 여러 가지 내장 애플릿이 포함되어 있습니다.

[[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

조건

if..에 대한 명령은 다음과 같습니다..그런 다음...else 연산자

```
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#
```

이러한 비교 연산자는 정수 값을 사용하는 데 사용됩니다.

연산자 설명

- eq Arg1-eq Arg2. Arg1이 Arg2와 같으면 True입니다.
- ne Arg1-ne Arg2. Arg1이 Arg2와 같지 않으면 True입니다.
- lt Arg1-lt Arg2. Arg1이 Arg2보다 작으면 True입니다.
- gt Arg1-gt Arg2. Arg1이 Arg2보다 큰 경우 True
- ge Arg1-ge Arg2. Arg1이 Arg2보다 크거나 같으면 true입니다.
- le Arg1-le Arg2. Arg1이 Arg2보다 작거나 같으면 True입니다.

다음 조건을 사용하여 파일을 사용할 수 있습니다.

연산자 설명

- a 또는 -e 파일이 있는 경우 True
- d 파일이 있고 디렉터리인 경우 True
- f 파일이 있고 일반 파일인 경우 True
- r 파일이 있고 읽을 수 있는 경우 True
- s 파일이 있고 크기가 0보다 큰 경우 True입니다.
- w 파일이 있고 실행 파일인 경우 True

-nt file1이 file2보다 최신인지 테스트합니다. 파일의 수정 날짜는 이 비교에 사용됩니다.
-발신 file1이 file2보다 오래되었는지 테스트

비교 연산자 내에서 여러 조건을 사용하려면 다음 논리 연산자를 사용할 수 있습니다.

&& Logical AND
|| Logical OR

루프

루프는 Linux 셸과 동일합니다. 두 개의 연산자를 사용할 수 있습니다.

for 연산자에 대한 명령은 다음과 같습니다.

```
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#
```

while 연산자에 대한 명령은 다음과 같습니다.

```
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#
```

문자열 작업

문자열 비교의 경우 if....그런 다음...else 연산자는 다음 수정 사항과 함께 사용할 수 있습니다.

연산자 설명

== 문자열 1 == string2, 문자열이 같으면 True입니다.
!= 문자열 1 != string2, 문자열이 같지 않으면 True
< string1 < string2, string1이 string2보다 사전적이면 True
> string1 > string2, string1이 string2보다 사전순으로 큰 경우 True

컨피그레이션 예시

```
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#
```

다음을 확인합니다.

이 섹션을 사용하여 컨피그레이션이 제대로 작동하는지 확인합니다.

현재 터미널 컨피그레이션을 표시하려면 **show terminal** 명령을 입력합니다.

```
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
```

사용자 정의 함수를 포함하는 모듈에서 사용 가능한 모든 기능을 표시하려면 **show shell functions** 명령을 입력합니다.

```
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
}
```

문제 해결

현재 이 컨피그레이션에 사용할 수 있는 특정 문제 해결 정보가 없습니다.

관련 정보

- [Catalyst 6500 컨피그레이션 가이드](#)
- [기술 지원 및 문서 - Cisco Systems](#)