

Exemplo de configuração do módulo CLI do Shell do Cisco IOS

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[Configurar](#)

[Tubos e redirecionamento](#)

[Miniaplicativos integrados](#)

[Condições](#)

[Circuitos](#)

[Operações de String](#)

[Exemplo de configuração](#)

[Verificar](#)

[Troubleshoot](#)

[Informações Relacionadas](#)

Introduction

Este documento descreve o módulo CLI do Cisco IOS[®] Shell (IOS.sh).

Prerequisites

Requirements

A Cisco recomenda que você tenha conhecimento do Cisco IOS versão 15.1(4)M, 15.1(2)S e versões posteriores. O Cisco IOS.sh deve ser configurado e ativado para usar os recursos e as funções do Cisco IOS.sh no roteador.

Componentes Utilizados

As informações neste documento são baseadas nas seguintes versões de hardware e software:

- Switches Cisco Catalyst 6500 Series

- Cisco IOS versão 15.1(1)SY

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. If your network is live, make sure that you understand the potential impact of any command.

Informações de Apoio

O Cisco IOS.sh é um módulo que permite usar scripts shell na CLI do Cisco IOS. Este módulo ajuda a automatizar ações de rotina e simplifica o gerenciamento de dispositivos. Ele pode ser usado em conjunto com outras ferramentas de gerenciamento, como o Cisco IOS Embedded Event Manager (EEM).

Configurar

Note: Use a [Command Lookup Tool \(somente clientes registrados\) para obter mais informações sobre os comandos usados nesta seção.](#)

Para habilitar a CLI do Shell no dispositivo, insira:

```
Switch#terminal shell
```

Para desabilitar a CLI do Shell, insira:

```
Switch#terminal no shell
```

Este recurso suporta a variável PATH, que contém uma lista de diretórios onde shell pode procurar funções/miniaplicativos. Além disso, a CLI do Shell suporta estes recursos:

Tubos e redirecionamento

```
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)
```

Miniaplicativos integrados

O Cisco IOS.sh contém vários miniaPLICATIVOS incorporados que ajudam a criar aplicativos e filtros para as funções do usuário:

```
[[          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
```

Condições

Aqui estão os comandos para o **se.....então....outro** operador.

```
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#
```

Estes operadores de comparação são usados para trabalhar com valores inteiros:

Operadores Descrição

-eq	Arg1-eq Arg2. Verdadeiro se Arg1 é igual a Arg2
-ne	Arg1-ne Arg2. Verdadeiro se Arg1 não é igual a Arg2
-lt	Arg1-lt Arg2. Verdadeiro se Arg1 é menor que Arg2
-gt	Arg1-gt Arg2. Verdadeiro se Arg1 é maior que Arg2
-ge	Arg1-ge Arg2. Verdadeiro se Arg1 é maior ou igual a Arg2
-le	Arg1-le Arg2. Verdadeiro se Arg1 é menor ou igual a Arg2

Essas condições estão disponíveis para trabalhar com arquivos:

Operadores Descrição

-a ou -e	Verdadeiro se o arquivo existir
-d	Verdadeiro se o arquivo existir e ele for um diretório
-f	Verdadeiro se o arquivo existe e é um arquivo normal

-r	Verdadeiro se o arquivo existe e é legível
-s	Verdadeiro se o arquivo existir e tiver um tamanho maior que zero
-w	Verdadeiro se o arquivo existe e é executável
-nt	Teste se o arquivo 1 é mais recente que o arquivo 2. A data de modificação no arquivo é usada para esta comparação
-t	Testar se o arquivo 1 é mais antigo que o arquivo 2

Para usar várias condições em um operador de comparação, você pode usar estes operadores lógicos:

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

Circuitos

Os loops são os mesmos do shell do Linux. Existem dois operadores disponíveis.

Estes são os comandos para o **para o** operador:

```
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#
```

Estes são os comandos para o operador **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#
```

Operações de String

Para comparações de strings, o **if.....então....else** operador pode ser usado com estas modificações:

Operadores Descrição

==	String 1 == string2, Verdadeiro se as strings forem iguais.
!=	String1 != string2 , Verdadeiro se as strings não forem iguais
<	String1 < string2, Verdadeiro se string1 é lexicograficamente menor que string2
>	String1 > string2, Verdadeiro se string1 é lexicograficamente maior que string2

Exemplo de configuração

```
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#

Verificar

Use esta seção para confirmar se a sua configuração funciona corretamente.

Insira o comando **show terminal** para exibir a configuração atual do terminal:

```
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
```

Insira o comando **show shell functions** para exibir todas as funções disponíveis no módulo, que incluem funções definidas pelo usuário:

```
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
}
```

Troubleshoot

Atualmente, não existem informações disponíveis específicas sobre Troubleshooting para esta configuração.

Informações Relacionadas

- [Guias de configuração do Catalyst 6500](#)
- [Suporte Técnico e Documentação - Cisco Systems](#)