

Exemple de configuration de module CLI de shell de Cisco IOS

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Informations générales](#)

[Configurez](#)

[Canaux et redirection](#)

[Applet encastés](#)

[Conditions](#)

[Boucles](#)

[Exécutions de chaîne](#)

[Exemple de configuration](#)

[Vérifiez](#)

[Dépannez](#)

[Informations connexes](#)

Introduction

Ce document décrit le module CLI du shell de Cisco IOS® (IOS.sh).

Conditions préalables

Conditions requises

Cisco recommande que vous ayez la connaissance de la Cisco IOS version 15.1(4)M, du 15.1(2)S, et des versions ultérieures. Cisco IOS.sh doit être configuré et activé afin d'utiliser les caractéristiques et les fonctions de Cisco IOS.sh sur votre routeur.

[Composants utilisés](#)

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Commutateurs de la gamme Cisco Catalyst 6500
- Cisco IOS version 15.1(1)SY

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

Informations générales

Cisco IOS.sh est un module qui te permet pour utiliser le script de shell dans le Cisco IOS CLI. Ce module aide à automatiser des actions courantes et simplifie la Gestion de périphériques. Il peut être utilisé en même temps que d'autres outils de gestion tels que le gestionnaire d'événement encastré par Cisco IOS (EEM).

Configurez

Remarque: Utilisez l'[Outil de recherche de commande](#) (clients [enregistrés](#) seulement) pour obtenir plus d'informations sur les commandes utilisées dans cette section.

Afin d'activer le shell CLI sur le périphérique, entrez :

```
Switch#terminal shell
```

Afin de désactiver le shell CLI, entrez :

```
Switch#terminal no shell
```

Ce prises en charge de fonctionnalité la variable de CHEMIN, qui contient une liste de répertoires où le shell peut rechercher la fonction/applet. En outre, le shell CLI prend en charge ces caractéristiques :

Canaux et redirection

```
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 encastrés

Cisco IOS.sh contient plusieurs applet intégrés qui aident à créer des demandes et des filtres de fonctions d'utilisateur :

```
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)
```

Conditions

Voici les commandes pour **si....puis....opérateur d'autre**.

```
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#
```

Ces opérateurs de comparaison sont utilisés afin de travailler avec des valeurs entières :

Opérateurs Description

- eq Arg1-eq Arg2. Rectifiez si Arg1 est égal à Arg2
- Ne Arg1-ne Arg2. Rectifiez si Arg1 n'est pas égal à Arg2
- lt Arg1-lt Arg2. Rectifiez si Arg1 est moins qu'Arg2
- gt Arg1-gt Arg2. Rectifiez si Arg1 est plus grand qu'Arg2
- GE Arg1-ge Arg2. Rectifiez si Arg1 est greate qu'ou égalez à Arg2
- le Arg1-le Arg2. Rectifiez si Arg1 est inférieur ou égal à Arg2

Ces conditions sont disponibles afin de fonctionner avec des fichiers :

Opérateurs Description

- a ou - e Rectifiez si le fichier existe
- d Rectifiez si le fichier existent et c'est un répertoire
- f Rectifiez si le fichier existe et est un fichier régulier
- r Rectifiez si le fichier existe et est accessible en lecture
- s Rectifiez si le fichier existe et a une taille plus grande que zéro
- W Rectifiez si le fichier existe et est exécutable
- NT Testez si file1 est plus nouveau que file2. La date de modification sur le fichier est utilisée pour cette comparaison
- ot Testez si file1 est plus ancien que file2

Afin d'utiliser plusieurs conditions chez un opérateur de comparaison, vous pouvez utiliser ces opérateurs logiques :

```
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#
```

Boucles

Les boucles sont identiques que dans le shell de Linux. Deux opérateurs sont disponibles.

Voici les commandes pour pour l'opérateur :

```
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#
```

Voici les commandes pour l'opérateur de moment :

```
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#
```

Exécutions de chaîne

Pour des comparaisons de chaîne, **si....puis....**l'opérateur **d'autre** peut être utilisé avec ces modifications :

Opérateurs Description

- == Ficelez 1 == string2, vrai si les chaînes sont égales.
- != String1 != string2, rectifient si les chaînes ne sont pas égales
- < String1 < string2, rectifient si string1 est lexicographique moins que string2
- > String1 > string2, rectifient si string1 est lexicographique plus grand que string2

Exemple de configuration

```
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#

```

Vérifiez

Utilisez cette section pour confirmer que votre configuration fonctionne correctement.

Sélectionnez la commande de **show terminal** afin d'afficher la configuration terminale en cours :

```

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

```

Écrivez les **fonctions de shell d'exposition** commandent afin d'afficher toutes les fonctions disponibles dans le module, qui incluent des fonctions définies par l'utilisateur :

```

Switch#show shell functions
#User defined functions:

Function namespace: DEFAULT
function add_desc()
{
show running-config interface Gi$1
figure terminal
interface Gi$1
description $2
end
show running-config interface Gi$1
}

```

Dépannez

Il n'existe actuellement aucune information de dépannage spécifique pour cette configuration.

Informations connexes

- [Guides de configuration de Catalyst 6500](#)
- [Support et documentation techniques - Cisco Systems](#)