

通过 SNMP 在 CatcOs 交换机和 TFTP 服务器之间移动文件和镜像

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[规则](#)

[复制从TFTP server的一配置到Catalyst交换机运行CatOS](#)

[逐步指导](#)

[检查结果](#)

[排除故障进程](#)

[复制从Catalyst交换机运行CatOS的一配置到TFTP server](#)

[逐步指导](#)

[验证进程](#)

[排除故障进程](#)

[复制从TFTP server的一个系统软件镜像到Catalyst交换机运行CatOS](#)

[逐步指导](#)

[验证进程](#)

[排除故障进程](#)

[复制从Catalyst交换机运行CatOS的一个系统软件镜像到TFTP server](#)

[逐步指导](#)

[验证进程](#)

[排除故障进程](#)

[UNIX 脚本示例](#)

[附录 A – MIB 对象详细资料](#)

[相关信息](#)

简介

本文描述如何移动配置文件和系统软件镜像在一交换机用Catalyst操作系统(CatOS)和在UNIX的一个简单文件传输协议(TFTP)服务器之间与简单网络管理协议(SNMP)。

先决条件

要求

验证您能ping TFTP server的IP地址从Catalyst交换机的：

```
Cat6509> (enable) ping 171.68.191.135
```

```
!!!!
```

```
----171.68.191.135 PING Statistics----
```

```
5 packets transmitted, 5 packets received, 0% packet loss
```

```
round-trip (ms)  min/avg/max = 2/2/2
```

这些步骤是：

- 不可适用为根据Cisco IOS软件的Catalyst交换机，例如Catalyst 2900/3500XL系列。
- 不可适用为有Cisco IOS软件的Catalyst 6000系列MSFC和MSFC2模块。
- 不可适用，如果SNMP读写团体串在交换机没有配置也不知道。参考[如何配置](#)详细的过程的[SNMP团体字符串](#)关于怎样配置SNMP团体字符串。
- 基于[NET-SNMP](#)（以前叫作UCD-SNMP）工具命令行语法。[如果有一些其他SNMP应用程序，例如HP OpenView或NetView，语法也许是与这些示例不同。](#)
- 凭[CISCO-STACK-MIB](#)，由Catalyst OS支持从初始Supervisor模块软件版本。参考在Cisco.com的[MIB支持的按产品](#)页验证您的交换机支持[CISCO-STACK-MIB](#)。使用从此MIB的这些MIB对象：

MIB对象对象名	OID
tftpHost	.1.3.6.1.4.1.9.5.1.5.1
tftpFile	.1.3.6.1.4.1.9.5.1.5.2
tftpModule	.1.3.6.1.4.1.9.5.1.5.3
tftpAction	.1.3.6.1.4.1.9.5.1.5.4
tftpResult	.1.3.6.1.4.1.9.5.1.5.5

请参阅[附录A](#)关于与定义的这些MIB对象的更多信息。

使用的组件

本文档中的信息根据运行仅Catalyst OS软件的交换机。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

规则

在所有示例中，这些值使用为说明：

- Catalyst 6509交换机用CatOS
- Catalyst 6509交换机的172.16.99.66 = IP地址
- **私有** = SNMP读写团体串。请使用在您的交换机配置的读写字符串。用**show snmp**命令验证此在交换机CLI。
- **公共** = SNMP只读属性字段。请使用在您的交换机配置的只读字符串。用**show snmp**命令验证此在交换机CLI。
- TFTP server的171.68.191.135 = IP地址

这是**snmpset**和**snmpwalk** in命令的语法这些示例：

```
snmpset [options...] <hostname> {<community>} [<objectID> <type> <value> ...] snmpwalk
```

[options...] <hostname> {<community>} [<objectID>]

有关文档规则的详细信息，请参阅 [Cisco 技术提示规则](#)。

复制从TFTP server的配置到Catalyst交换机运行CatOS

这些步骤通过进程指导您复制配置文件。

逐步指导

完成这些步骤：

1. 创建一个新的文件， **switch-config**，在TFTP server /tftpboot目录。在UNIX，请使用此语法：**涉及<filename>**。

```
touch switch-config
```

2. 更改文件的权限对**777**。使用以下语法：**chmod <permissions> <filename>**。

```
chmod 777 switch-config
```

3. 定义TFTP server的IP地址与**tftpHost** MIB对象的：

```
chmod 777 switch-config
```

4. 定义您将使用复制配置的TFTP文件名，与**tftpFile** MIB对象：

```
chmod 777 switch-config
```

5. 选择在配置将传送的Catalyst交换机的模块，有**tftpModule** MIB对象的。选择Supervisor模块而不是MSFC或MSFC2模块，否则发生故障。验证**snmpset**命令的正确模块号用一**show module**命令在交换机CLI。典型输出是：

Mod	Slot	Ports	Module-Type	Model	Sub	Status
2	2	2	1000BaseX Supervisor	WS-X6K-SUP1A-2GE	yes	ok
16	2	1	Multilayer Switch Feature	WS-F6K-MSFC	no	OK
....						

--<snip>--

在输出示例:中， Supervisor模块数是2并且在插槽编号2。使用2为了定义**tftpModule** MIB对象：

Mod	Slot	Ports	Module-Type	Model	Sub	Status
2	2	2	1000BaseX Supervisor	WS-X6K-SUP1A-2GE	yes	ok
16	2	1	Multilayer Switch Feature	WS-F6K-MSFC	no	OK
....						

--<snip>--

6. 请使用**tftpAction** MIB对象为了定义将从TFTP server转接到有MIB对象值的交换机为**2 = downloadConfig**的交换机配置文件。请参阅在[附录A](#)的MIB对象详细资料：

Mod	Slot	Ports	Module-Type	Model	Sub	Status
2	2	2	1000BaseX Supervisor	WS-X6K-SUP1A-2GE	yes	ok

```
16 2 1 Multilayer Switch Feature WS-F6K-MSFC no OK
....
--<snip>--
```

检查结果

为了验证这些操作结果，请执行这些步骤之一：

1. 轮询`tfTtpGrp (.1.3.6.1.4.1.9.5.1.5)` MIB对象并且比较与[附录A](#)的结果：

```
Mod Slot Ports Module-Type Model Sub Status
--- --- ---
2 2 2 1000BaseX Supervisor WS-X6K-SUP1A-2GE yes ok
16 2 1 Multilayer Switch Feature WS-F6K-MSFC no OK
....
--<snip>--
```

2. 轮询`tfTtpResult` MIB对象并且比较与MIB对象详细资料的输出在[附录A](#)：

```
Mod Slot Ports Module-Type Model Sub Status
--- --- ---
2 2 2 1000BaseX Supervisor WS-X6K-SUP1A-2GE yes ok
16 2 1 Multilayer Switch Feature WS-F6K-MSFC no OK
....
--<snip>--
```

排除故障进程

如果下载是成功的，MIB对象输出与2 (或成功)是相等的。如果收到若干其他输出，它与`TFTPResult`对象的[附录A比较](#)并且采取适当的步骤。

复制从Catalyst交换机运行CatOS的配置到TFTP server

这些步骤通过进程指导您复制配置文件。

逐步指导

完成这些步骤：

1. 创建一个新的文件，`switch-config`，在TFTP server `/tftpboot`目录。在UNIX，请使用此语法：`涉及<filename>`。

```
touch switch-config
```

2. 更改文件的权限对777有此语法的：`chmod <permissions> <filename>`。

```
chmod 777 switch-config
```

3. 定义TFTP server的IP地址与`tfTtpHost` MIB对象的。语法为：

```
chmod 777 switch-config
```

4. 定义您将使用复制配置的TFTP文件名，与`tfTtpFile` MIB对象：

```
chmod 777 switch-config
```

5. 选择在配置将传送的Catalyst交换机的模块，有tftpModule MIB对象的。选择Supervisor模块而不是MSFC或MSFC2模块，否则发生故障。验证snmpset命令的正确模块号用—show module命令在交换机CLI。典型输出是：

```

Mod Slot  Ports  Module-Type                Model                Sub  Status
---  ---  ---  ---  ---  ---
2    2    2    1000BaseX Supervisor      WS-X6K-SUP1A-2GE    yes  ok
16   2    1    Multilayer Switch Feature  WS-F6K-MSFC         no   OK
....
--<snip>--

```

在输出示例:中， Supervisor模块数是2并且在插槽编号2。使用2为了定义tftpModule MIB对象：

```

Mod Slot  Ports  Module-Type                Model                Sub  Status
---  ---  ---  ---  ---  ---
2    2    2    1000BaseX Supervisor      WS-X6K-SUP1A-2GE    yes  ok
16   2    1    Multilayer Switch Feature  WS-F6K-MSFC         no   OK
....
--<snip>--

```

6. 请使用tftpAction MIB对象为了定义交换机配置文件将从TFTP server转接到有MIB对象值的交换机为3 = uploadConfig。请参阅在[附录A](#)的MIB对象详细资料：

```

Mod Slot  Ports  Module-Type                Model                Sub  Status
---  ---  ---  ---  ---  ---
2    2    2    1000BaseX Supervisor      WS-X6K-SUP1A-2GE    yes  ok
16   2    1    Multilayer Switch Feature  WS-F6K-MSFC         no   OK
....
--<snip>--

```

验证进程

为了验证这些操作结果，请执行这些步骤之一：

1. 轮询tftpGrp (.1.3.6.1.4.1.9.5.1.5) MIB对象并且比较与[附录A](#)的结果：

```

Mod Slot  Ports  Module-Type                Model                Sub  Status
---  ---  ---  ---  ---  ---
2    2    2    1000BaseX Supervisor      WS-X6K-SUP1A-2GE    yes  ok
16   2    1    Multilayer Switch Feature  WS-F6K-MSFC         no   OK
....
--<snip>--

```

2. 轮询tftpResult MIB对象并且比较与MIB对象详细资料的输出在[附录A](#)：

```

Mod Slot  Ports  Module-Type                Model                Sub  Status
---  ---  ---  ---  ---  ---
2    2    2    1000BaseX Supervisor      WS-X6K-SUP1A-2GE    yes  ok
16   2    1    Multilayer Switch Feature  WS-F6K-MSFC         no   OK
....
--<snip>--

```

排除故障进程

如果下载是成功的，MIB对象输出与2 (或成功)是相等的。如果收到若干其他输出，它与TFTPResult对象的[附录A比较](#)并且采取适当的步骤。

注意： 此步骤如在输出show config all命令中看到转接默认和非默认配置从交换机，在交换机CLI在

特权模式。show config命令在交换机显示仅非默认配置。

复制从TFTP server的系统软件镜像到Catalyst交换机运行CatOS

这些步骤通过进程指导您复制软件镜像。

逐步指导

完成这些步骤：

1. 下载并且安置正确Supervisor镜像镜像文件在/tftpboot目录在TFTP server。在本例中，**cat6000-sup.5-4-2a.bin**使用为说明。
2. 更改文件的权限对777有此语法的：**chmod <permissions> <filename>**。

```
chmod 777 cat6000-sup.5-4-2a.bin
```

3. 定义使用tftpHost MIB对象TFTP server的IP地址：

```
chmod 777 cat6000-sup.5-4-2a.bin
```

4. 定义您将使用复制镜像文件的TFTP文件名：

```
chmod 777 cat6000-sup.5-4-2a.bin
```

5. 在本例中，Supervisor模块数是2并且在插槽编号2如在show module命令输出中看到。使用2为了定义tftpModule MIB对象：

```
chmod 777 cat6000-sup.5-4-2a.bin
```

这意味着CatOS镜像现在TFTP server的/tftpboot目录转接对Supervisor模块闪存如在输出show flash命令中看到。

6. 请使用tftpAction MIB对象为了定义镜像文件从TFTP server转接到有MIB对象值的交换机为4 = downloadSw。请参阅在[附录A](#)的MIB对象详细资料：

```
chmod 777 cat6000-sup.5-4-2a.bin
```

验证进程

为了验证结果此操作，执行这些步骤之一：

1. 轮询tftpGrp (.1.3.6.1.4.1.9.5.1.5) MIB对象并且比较与[附录A](#)的结果：

```
chmod 777 cat6000-sup.5-4-2a.bin
```

注意：最后一项显示图像传送在进程。等待几分钟，然后再轮询tftpResult MIB对象为了验证

顺利地转接。此步骤能花费几分钟完成，取决于镜像文件大小(字节)。当镜像传输过程进行中时，如果发出**show flash**命令在交换机，您将看到：

```
chmod 777 cat6000-sup.5-4-2a.bin
```

2. 轮询**tftpResult** MIB对象并且比较与MIB对象详细资料的输出在[附录A](#)：

```
chmod 777 cat6000-sup.5-4-2a.bin
```

[排除故障进程](#)

如果下载是成功的，MIB对象输出与2 (或成功)是相等的。如果收到其他输出，它与**TFTPResult**对象的[附录A比较](#)并且采取适当的步骤。

一旦图像传送成功地完成，请验证镜像文件大小(字节)匹配在**show flash**命令输出中显示的那对在TFTP server (cat6000-sup.5-4-2a.bin的文件，在本例中)。

[复制从Catalyst交换机运行CatOS的系统软件镜像到TFTP server](#)

这些步骤通过进程指导您复制软件镜像。

[逐步指导](#)

完成这些步骤：

1. 创建在TFTP server的/tftpboot目录的一新的文件**image.bin**。在UNIX，请使用此语法：涉及**<filename>**。请使用**.bin**as文件扩展。

```
touch image.bin
```

2. 更改文件的权限对**777**有语法的：**chmod <permissions> <filename>**。

```
chmod 777 image.bin
```

3. 使用**tftpHost** MIB对象，定义TFTP server的IP地址：

```
chmod 777 image.bin
```

4. 定义您将使用复制有**tftpFile** MIB对象的镜像文件的TFTP文件名：

```
chmod 777 image.bin
```

5. 在本例中，Supervisor模块数是2并且在插槽编号2如在**show module**命令输出中看到。使用2为了定义**tftpModule** MIB对象：

```
chmod 777 image.bin
```

这意味着在闪存的Supervisor模块运行的CatOS镜像转接对TFTP server如在输出**show flash**命

令中看到。

6. 请使用 `tftpAction` MIB对象为了定义镜像文件从TFTP server转接到有MIB对象值的交换机为 `5 = uploadSw`。请参阅在[附录A](#)的MIB对象详细资料：

```
chmod 777 image.bin
```

验证进程

为了验证这些操作结果，请执行这些步骤之一：

1. 轮询 `tftpGrp (.1.3.6.1.4.1.9.5.1.5)` MIB对象并且比较与[附录A](#)的结果：

```
chmod 777 image.bin
```

注意：最后一项显示图像传送在进程。等待几分钟再然后轮询 `tftpResult` MIB对象为了验证顺利地转接。此步骤能花费几分钟完成，取决于镜像文件大小(字节)。

2. 轮询 `tftpResult` MIB对象并且比较与MIB对象详细资料的输出在[附录A](#)：

```
chmod 777 image.bin
```

排除故障进程

如果下载是成功的，MIB对象输出与2 (或成功)是相等的。如果收到其他输出，它与 `TFTPResult`对象的[附录A比较](#)并且采取适当的步骤。

一旦图像传送成功地完成，请验证在对文件的 `show flash`命令输出中显示在TFTP server的镜像文件大小(字节)匹配(`image.bin`，在本例中)。

注意：如果有多重镜像在闪存(`show flash`)，只有Supervisor模块启动的镜像，转接对与此步骤的TFTP server。请使用 `show boot`命令发现引导变量=，显示Supervisor模块用于从闪存的什么镜像启动。参考在[Catalyst交换机的配置文件升级软件镜像](#)欲知更多信息。

UNIX 脚本示例

注意：提供作为仅示例和不支持这些脚本由Cisco系统。

写脚本自动化在交换机的配置文件和Cisco IOS迁移

```
#!/bin/sh
# Script to automate config file & IOS migration of switches
# supporting STACK-MIB including 5000, 5500, 1400, 2900, 1200
if [ ! -f SW ] ;
then
echo
echo "File SW does not exist!!!"
echo
echo "Syntax is 'switch.sh'"
echo "where each line in file SW lists:"
echo "Switchname Filename Serverip Module# Moduleaction Community"
echo
```



```

echo "Switchname must resolve"
echo "Filename must exist in server tftpboot directory 777"
echo "Serverip is the ip of the server for the file"
echo "Module# is usually '1'"
echo "Module action is as per STACK-MIB: "
echo "- 2 - config file - server > switch"
echo "- 3 - config file - switch > server"
echo "- 4 - software image - server > switch"
echo "- 5 - software image - switch > server"
echo "Community is *write* community"
echo
exit
fi
cat SW |
while read SW
do
    SWNAME=\Qecho $SW | cut -d' ' -f 1\Q
    FILE=\Qecho $SW | cut -d' ' -f 2\Q
    SERVER=\Qecho $SW | cut -d' ' -f 3\Q
    MODULE=\Qecho $SW | cut -d' ' -f 4\Q
    ACTION=\Qecho $SW | cut -d' ' -f 5\Q
    CMTY=\Qecho $SW | cut -d' ' -f 6\Q
    echo
    echo $SWNAME
    echo $FILE
    echo $SERVER
    echo $MODULE
    echo $ACTION
    echo $CMTY
    echo
    # '-t #' can be modified to adjust timeout
    snmpset -t 100 -c $CMTY $SWNAME .1.3.6.1.4.1.9.5.1.5.1.0 octetstring $SERVER
    sleep 5
    snmpset -t 100 -c $CMTY $SWNAME .1.3.6.1.4.1.9.5.1.5.2.0 octetstring $FILE
    sleep 5
    snmpset -t 100 -c $CMTY $SWNAME .1.3.6.1.4.1.9.5.1.5.3.0 integer $MODULE
    sleep 5
    snmpset -t 100 -c $CMTY $SWNAME .1.3.6.1.4.1.9.5.1.5.4.0 integer $ACTION
sleep 60
echo
echo Check Progress...
echo
echo
echo "Switch $SWNAME: \\c"; snmpget -t 100 -c $CMTY $SWNAME .1.3.6.1.4.1.9.5.1.5.5.0 | cut
-d": " -f 3
done

```

交换Expect脚本执行在交换机的一个特定命令

```

#!/usr/nms/bin/expect
# Above line points to your expect interpreter
# Add '-d' option to expect line above to enable debugging
# Tested on Cat5000 with regular login; no error-checking
# except for number arguments, but will timeout on failure.
# Tacacs+ lines left in for future releases
set argc [llength $argv]
if { $argc < 4 } {
    puts "Syntax is:"
    puts "(For system with no Tac+)"
    puts "switch.exp destination \"command\" vtypassword enapassword"
    exit 0 }
set destination [lindex $argv 0]
puts -nonewline "Where we're going: "

```

```

puts $destination
set command [lindex $argv 1]
puts -nonewline "What we're doing:  "
puts $command
set vtypassword [lindex $argv 2]
puts -nonewline "What our password is (vty):  "
puts $vtypassword
set enapassword [lindex $argv 3]
puts -nonewline "What our password is (enable):  "
puts $enapassword
# username only for Tac+
set username [lindex $argv 4]
puts -nonewline "What our username is if Tac+:  "
puts $username
#
set timeout 10
spawn telnet $destination
expect {
    "Enter password:" {
        send "$vtypassword\r" }
    "Username:" {
        send "$username\r"
        exec sleep 1
        expect "Password:"
        send "$vtypassword\r"
    }
}
# Look for non-enable router 'prompt>'
expect -re "(^.*)(\r\n\[^\ ]+> \$)"
# Get into enable mode
send "en\r"
expect {
    "password: " {
        send "$enapassword\r" }
    "Username:" {
        send "$username\r"
        exec sleep 1
        expect "Password:"
        send "$enapassword\r" }
}
# Look for enable router 'prompt#'
expect -re "(^.*)(\r\n\[^\ ]+(enable) \$)"
# Send the command
send "$command\r"
expect {
    -re "(^.*)(\r\n\[^\ ]+ (enable) \$)" {
        append buffer $expect_out(1,string)
    }
    -re "(^.*)(\r\n\ --More-- \$)" {
        append buffer $expect_out(1,string)
        send " "
    }
    -re "(^.*)(\r\n\ --More-- \$)" {
        append buffer $expect_out(1,string)
        send " "
    }
}
# Done with command - disable prior to exit
send "disable\r"
expect -re "(^.*)(\r\n\[^\ ]+> \$)"
exec sleep 1
send "logout"

```

显示通过SNMP同一输出的Perl脚本作为“show cam dynamic”

OID	.1.3.6.1.4.1.9.5.1.5.3
类型	整数
权限	读写
状态	当前
范围	0 - 16
MIB	CISC O-STACK-MIB
说明	模块的哪代码/配置转接。
在树的OID	: := {Iso(1) org(3) dod(6) internet(1) private(4) enterprises(1) cisco(9) workgroup(5) ciscoStackMIB(1) tftpGrp(5) 3}

对象	tftpAction
OID	.1.3.6.1.4.1.9.5.1.5.4
类型	整数
权限	读写
状态	当前
值	<ol style="list-style-type: none"> 1. 其他 2. downloadConfig 3. uploadConfig 4. downloadSw 5. uploadSw 6. downloadFw 7. uploadFw
MIB	CISCO-STACK-MIB
说明	如果集此对象到其中一个可接受值，它启动与在 tftpHost给的信息的请求的操作， tftpFile， tftpModule。downloadConfig(2)：从主机/文件 uploadConfig(3)的接收配置：发送配置主机的/文件 downloadSw(4)：从主机/文件uploadSw(5)的接收软件镜像：发送软件镜像主机的/文件 downloadFw(6)：从主机/文件uploadFw(7)的接收固件镜像：发送固件镜像主机/文件，如果集对其他值的此对象，您收到错误。
在树的OID	: := {Iso(1) org(3) dod(6) internet(1) private(4) enterprises(1) cisco(9) workgroup(5) ciscoStackMIB(1) tftpGrp(5) 4}

对象	tftpResult
OID	.1.3.6.1.4.1.9.5.1.5.5
类型	整数
权限	只读
状态	当前
值	1. inProgress

	2. 成功 3. noResponse 4. tooManyRetries 5. 无缓冲 6. noProcesses 7. badChecksum 8. badLength 9. badFlash 10. serverError 11. userCanceled 12. wrongCode 13. fileNotFound 14. invalidTftpHost 15. invalidTftpModule 16. accessViolation 17. unknownStatus 18. invalidStorageDevice 19. insufficientSpaceOnStorageDevice 20. insufficientDramSize 21. incompatibleImage
MIB	CISCO-STACK-MIB
说明	包含最后TFTP行动请求的结果
在树的OID	: : = {iso(1) org(3) dod(6) internet(1) private(4) enterprises(1) cisco(9) workgroup(5) ciscoStackMIB(1) tftpGrp(5) 5}

相关信息

- [通过 SNMP 在路由器和 TFTP 服务器之间移动文件和镜像](#)
- [思科MIB下载](#)
- [技术支持和文档 - Cisco Systems](#)