Moving Files and Images Between a CatOS Switch and TFTP Server via SNMP

Document ID: 13504

Contents

Introduction

This document describes how to move configuration files and system software images between a switch with the Catalyst Operating Systems (CatOS) and a Trivial File Transfer Protocol (TFTP) server on UNIX with the Simple Network Management Protocol (SNMP).

Prerequisites

Requirements

Verify that you can ping the IP address of the TFTP server from the Catalyst switch:

```
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
```

These procedures are:

- Not applicable for Catalyst switches based on Cisco IOS® software, such as the Catalyst 2900/3500XL series.
- Not applicable for Catalyst 6000 series MSFC and MSFC2 modules with Cisco IOS software.
- Not applicable if SNMP Read−Write community string is not configured or known on the switch. Refer to How to Configure SNMP Community Strings for the detailed procedure on how to configure the SNMP community strings.
- Based on command line syntax of NET−SNMP ⬈ (previously known as UCD−SNMP) Utilities. If you have some other SNMP applications, such as HP Open View or NetView, the syntax might be different from these examples.
- Based on CISCO−STACK−MIB, which is supported by the Catalyst OS since the initial supervisor module software version. Refer to the MIBs Supported by Product page on Cisco.com to verify that your switch supports CISCO−STACK−MIB. These MIB objects from this MIB are used:

MIB Object Name	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

See Appendix A for more information on these MIB objects with definitions.

Components Used

The information in this document is based on the switches that run only Catalyst OS software.

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.

Conventions

In all the examples, these values are used for illustration:

- Catalyst 6509 Switch with CatOS
- **172.16.99.66** = IP address of the Catalyst 6509 Switch
- **private** = SNMP read−write community string. Use the read−write string configured on your switch. Verify this with the **show snmp** command on the switch CLI.
- **public** = SNMP read−only community string. Use the read−only string configured on your switch. Verify this with the **show snmp** command on the switch CLI.
- **171.68.191.135** = IP address of the TFTP server

This is the syntax for the **snmpset** and **snmpwalk** commands in these examples:

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

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Copy a Configuration from the TFTP Server to the Catalyst Switch Running CatOS

These steps guide you through the process to copy a configuration file.

Step–by–Step Instructions

Complete these steps:

1. Create a new file, **switch–config**, in the TFTP server **/tftpboot** directory. On UNIX, use this syntax: **touch** *<filename>*.

   ```
   touch switch-config
   ```
2. Change the permissions of the file to **777**. Use this syntax: **chmod** *<permissions>* *<filename>* .

   ```
   chmod 777 switch-config
   ```
3. Define the IP address of the TFTP server with the **tftpHost** MIB object:

   ```
   % snmpset 172.16.99.66 private .1.3.6.1.4.1.9.5.1.5.1.0 s 171.68.191.135
   enterprises.9.5.1.5.1.0 = "171.68.191.135"
   ```
4. Define the TFTP file name that you will use to copy the configuration, with the **tftpFile** MIB object:

   ```
   % snmpset 172.16.99.66 private .1.3.6.1.4.1.9.5.1.5.2.0 s switch-config
   enterprises.9.5.1.5.2.0 = switch-config
   ```
5. Select the module on the Catalyst switch where the configuration will be delivered, with the **tftpModule** MIB object. Select the Supervisor module and not MSFC or MSFC2 module, otherwise it fails. Verify the correct module number for the **snmpset** command with a **show module** command on the switch CLI. A typical output is:

   ```
   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>--
   ```

   In the sample output, the Supervisor module number is 2 and is in slot number 2. Use **2** in order to define the **tftpModule** MIB object:

   ```
   % snmpset 172.16.99.66 private .1.3.6.1.4.1.9.5.1.5.3.0 i 2
   enterprises.9.5.1.5.3.0 = 2
   ```
6. Use the **tftpAction** MIB object in order to define the switch configuration file that is to be transferred from the TFTP server to the switch with the MIB object value of **2 = downloadConfig**. See the MIB object details in Appendix A:

   ```
   % snmpset 172.16.99.66 private .1.3.6.1.4.1.9.5.1.5.4.0 i 2
   enterprises.9.5.1.5.4.0 = 2
   ```

Verify the Results

In order to verify the results of these operations, perform one of these steps:

1. Poll the **tftpGrp (.1.3.6.1.4.1.9.5.1.5)** MIB object and compare the results with Appendix A:

   ```
   % snmpwalk 172.16.99.66 public .1.3.6.1.4.1.9.5.1.5
   enterprises.9.5.1.5.1.0 = "171.68.191.135"
   ```

```
enterprises.9.5.1.5.2.0 = "switch-config"
```

```
enterprises.9.5.1.5.3.0 = 2
```

```
enterprises.9.5.1.5.4.0 = 2
```

```
enterprises.9.5.1.5.5.0 = 2
```

2. Poll the **tftpResult** MIB object and compare the output with the MIB object details in Appendix A:

```
% snmpwalk 172.16.99.66 public .1.3.6.1.4.1.9.5.1.5.5
enterprises.9.5.1.5.5.0 = 2
```

Troubleshoot the Process

If download is successful, the MIB object output is equal to 2 (or success). If you receive some other output, compare it with Appendix A for the **tftpResult** object and take appropriate steps.

Copy a Configuration from the Catalyst Switch Running CatOS to the TFTP Server

These steps guide you through the process to copy a configuration file.

Step−by−Step Instructions

Complete these steps:

1. Create a new file, **switch−config**, in the TFTP server **/tftpboot** directory. On UNIX, use this syntax: **touch** *<filename>*.

   ```
   touch switch-config
   ```
2. Change the permissions of the file to **777** with this syntax: **chmod** *<permissions> <filename>* .

   ```
   chmod 777 switch-config
   ```
3. Define the IP address of the TFTP server with the **tftpHost** MIB object. The syntax is:

   ```
   % snmpset 172.16.99.66 private .1.3.6.1.4.1.9.5.1.5.1.0 s 171.68.191.135
   enterprises.9.5.1.5.1.0 = "171.68.191.135"
   ```
4. Define the TFTP file name that you will use to copy the configuration, with the **tftpFile** MIB object:

   ```
   % snmpset 172.16.99.66 private .1.3.6.1.4.1.9.5.1.5.2.0 s switch-config
   enterprises.9.5.1.5.2.0 = switch-config
   ```
5. Select the module on the Catalyst switch where the configuration will be delivered, with the **tftpModule** MIB object. Select the Supervisor module and not MSFC or MSFC2 module, otherwise it

fails. Verify the correct module number for the **snmpset** command with a **show module** command on the switch CLI. A typical output is:

```
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>--
```

In the sample output, the Supervisor module number is 2 and is in slot number 2. Use 2 in order to define the **tftpModule** MIB object:

```
% snmpset 172.16.99.66 private .1.3.6.1.4.1.9.5.1.5.3.0 i 2
enterprises.9.5.1.5.3.0 = 2
```

6. Use the **tftpAction** MIB object in order to define that the switch configuration file is to be transferred from the TFTP server to the switch with the MIB object value of **3 = uploadConfig**. See the MIB object details in Appendix A:

```
% snmpset 172.16.99.66 private .1.3.6.1.4.1.9.5.1.5.4.0 i 3
enterprises.9.5.1.5.4.0 = 3
```

Verify the Process

In order to verify the results of these operations, perform one of these steps:

1. Poll the **tftpGrp (.1.3.6.1.4.1.9.5.1.5)** MIB object and compare the results with Appendix A:

```
% snmpwalk 172.16.99.66 public .1.3.6.1.4.1.9.5.1.5
enterprises.9.5.1.5.1.0 = "171.68.191.135"

!--- IP address of the TFTP server

enterprises.9.5.1.5.2.0 = "switch-config"

!--- name of the switch configuration file

enterprises.9.5.1.5.3.0 = 2

!--- Module number. In this case, Supervisor module

enterprises.9.5.1.5.4.0 = 1

!--- TFTP action

enterprises.9.5.1.5.5.0 = 2

!--- Result of the TFTP action, 2 = Succes
```

2. Poll the **tftpResult** MIB object and compare the output with the MIB object details in Appendix A:

```
% snmpwalk 172.16.99.66 public .1.3.6.1.4.1.9.5.1.5.5
enterprises.9.5.1.5.5.0= 2

!--- Result of the TFTP action, 2 = Success
```

Troubleshoot the Process

If download is successful, the MIB object output is equal to 2 (or success). If you receive some other output, compare it with Appendix A for the **tftpResult** object and take appropriate steps.

Note: This procedure transfers both default and non−default configurations from the switch, as seen in the output of the **show config all** command on the switch CLI in enable mode. The **show config** command on the switch shows only the non−default configurations.

Copy a System Software Image from the TFTP Server to the Catalyst Switch Running CatOS

These steps guide you through the process to copy a software image.

Step−by−Step Instructions

Complete these steps:

1. Download and place the correct Supervisor image file in the **/tftpboot** directory on the TFTP server. In this example, **cat6000−sup.5−4−2a.bin** is used for illustration.
2. Change the permissions of the file to **777** with this syntax: **chmod** *<permissions> <filename>*.

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

3. Define the IP address of the TFTP server that uses the **tftpHost** MIB object:

   ```
   % snmpset 172.16.99.66 private .1.3.6.1.4.1.9.5.1.5.1.0 s 171.68.191.135
   enterprises.9.5.1.5.1.0 = "171.68.191.135"
   ```
4. Define the TFTP file name that you will use to copy the image file:

   ```
   % snmpset 172.16.99.66 private .1.3.6.1.4.1.9.5.1.5.2.0 s cat6000-sup.5-4-2a.bin
   enterprises.9.5.1.5.2.0 = "cat6000-sup.5-4-2a.bin"
   ```
5. In this example, the Supervisor module number is 2 and is in slot number 2 as seen in the **show module** command output. Use **2** in order to define the **tftpModule** MIB object:

   ```
   % snmpset 172.16.99.66 private .1.3.6.1.4.1.9.5.1.5.3.0 i 2
   enterprises.9.5.1.5.3.0 = 2
   ```

   This means that the CatOS image present in the **/tftpboot** directory on the TFTP server is transferred to the Supervisor module flash as seen in the output of the **show flash** command.
6. Use the **tftpAction** MIB object in order to define that the image file is transferred from the TFTP server to the switch with the MIB object value of **4 = downloadSw**. See the MIB object details in Appendix A:

   ```
   % snmpset 172.16.99.66 private .1.3.6.1.4.1.9.5.1.5.4.0 i 4
   enterprises.9.5.1.5.4.0 = 4
   ```

Verify the Process

In order to verify the results of this operations, perform one of these steps:

1. Poll the **tftpGrp (.1.3.6.1.4.1.9.5.1.5)** MIB object and compare the results with Appendix A:

   ```
   % snmpwalk 172.16.99.66 public .1.3.6.1.4.1.9.5.1.5
   ```

```
enterprises.9.5.1.5.1.0 = "171.68.191.135"

!--- IP address of the TFTP server

enterprises.9.5.1.5.2.0 = "cat6000-sup.5-4-2a.bin"

!--- name of the switch image file

enterprises.9.5.1.5.3.0 = 0
enterprises.9.5.1.5.4.0 = 4

!--- TFTP action, 4 = downloadSw

enterprises.9.5.1.5.5.0 = 1

!--- Result of the TFTP action, 1 = In Process
```

Note: The last entry shows that the image transfer is in process. Wait a few minutes, then poll the **tftpResult** MIB object again in order to verify that it has transferred successfully. This step can take a few minutes to complete, which depends on the image file size (bytes). While the image transfer process is underway, if you issue a **show flash** command on the switch, you will see:

```
Cat6509> (enable) show flash
TFTP session in progress. Try again later.
```

2. Poll the **tftpResult** MIB object and compare the output with the MIB object details in Appendix A:

```
% snmpwalk 172.16.99.66 public .1.3.6.1.4.1.9.5.1.5.5
enterprises.9.5.1.5.5.0 = 2

!--- Result of the TFTP action, 2 = Success
```

Troubleshoot the Process

If download is successful, the MIB object output is equal to 2 (or success). If you receive other output, compare it with Appendix A for the **tftpResult** object and take appropriate steps.

Once the image transfer completes successfully, verify that the image file size (bytes) matches that shown in the **show flash** command output to the file in the TFTP server (cat6000−sup.5−4−2a.bin, in this example).

Copy a System Software Image from the Catalyst Switch Running CatOS to the TFTP Server

These steps guide you through the process to copy a software image.

Step−by−Step Instructions

Complete these steps:

1. Create a new file **image.bin** in the **/tftpboot** directory of the TFTP server. On UNIX, use this syntax: **touch** *<filename>*. Use **.bin** as the file extension.

   ```
   touch image.bin
   ```

2. Change the permissions of the file to **777** with the syntax: **chmod** *<permissions> <filename>*.

   ```
   chmod 777 image.bin
   ```

3. Define the IP address of the TFTP server using the **tftpHost** MIB object :

```
% snmpset 172.16.99.66 private .1.3.6.1.4.1.9.5.1.5.1.0 s 171.68.191.135
enterprises.9.5.1.5.1.0 = "171.68.191.135"
```
4. Define the TFTP file name you will use to copy the image file with the **tftpFile** MIB object :

```
% snmpset 172.16.99.66 private .1.3.6.1.4.1.9.5.1.5.2.0 s image.bin
enterprises.9.5.1.5.2.0 = "image.bin"
```
5. In this example, the Supervisor module number is 2 and is in slot number 2 as seen in the **show module** command output. Use 2 in order to define the **tftpModule** MIB object:

```
% snmpset 172.16.99.66 private .1.3.6.1.4.1.9.5.1.5.3.0 i 2
enterprises.9.5.1.5.3.0 = 2
```

This means that the CatOS image that runs on the Supervisor module in Flash is transferred to the TFTP server as seen in the output of the **show flash** command.
6. Use the **tftpAction** MIB object in order to define that the image file is transferred from the TFTP server to the switch with the MIB object value of **5 = uploadSw**. See the MIB object details in Appendix A:

```
% snmpset 172.16.99.66 private .1.3.6.1.4.1.9.5.1.5.4.0 i 5
enterprises.9.5.1.5.4.0 = 5
```

Verify the Process

In order to verify the results of these operations, perform one of these steps:

1. Poll the **tftpGrp (.1.3.6.1.4.1.9.5.1.5)** MIB object and compare the results with Appendix A:

```
% snmpwalk 172.16.99.66 public .1.3.6.1.4.1.9.5.1.5
enterprises.9.5.1.5.1.0 = "171.68.191.135"

!--- IP address of the TFTP server

enterprises.9.5.1.5.2.0 = "image.bin"

!--- name of the switch image file

enterprises.9.5.1.5.3.0 = 2

!--- Module number. In this case, Supervisor module

enterprises.9.5.1.5.4.0 = 5

!--- TFTP action, 5 = uploadSw

enterprises.9.5.1.5.5.0 = 1

!--- Result of the TFTP action, 1 = In Process
```

Note: The last entry shows that the image transfer is in process. Wait for a few minutes and then poll the **tftpResult** MIB object again in order to verify that it has transferred successfully. This step can take a few minutes to complete, which depends on the image file size (bytes).
2. Poll the **tftpResult** MIB object and compare the output with the MIB object details in Appendix A:

```
% snmpwalk 172.16.99.66 public .1.3.6.1.4.1.9.5.1.5.5
enterprises.9.5.1.5.5.0 = 2 --> Result of the TFTP action, 2 = Success
```

Troubleshoot the Process

If download is successful, the MIB object output is equal to 2 (or success). If you receive other output, compare it with Appendix A for the **tftpResult** object and take appropriate steps.

Once the image transfer completes successfully, verify the image file size (bytes) matches that are shown in the **show flash** command output to the file in the TFTP server (**image.bin**, in this example).

Note: If you have multiple images in the flash (**show flash**), only the image from which the Supervisor module has been booted up, is transferred to the TFTP server with this procedure. Use the **show boot** command to see the BOOT variable =, which shows what image from the flash is used by the Supervisor module to boot up. Refer to Upgrading Software Images and Working with Configuration Files on Catalyst Switches for more information.

UNIX Script Example

Note: These scripts are provided as examples only and are not supported in any way by Cisco Systems.

Script to Automate Configuration File and Cisco IOS Migration on Switches

```
#!/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
```

Switch Expect Script to Execute a Particular Command on the Switch

```
#!/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"
```

Perl Script to Show via SNMP the Same Output as "show cam dynamic"

```perl
#!/usr/local/bin/perl
open(TABLE, "bridge-table.csv") || die "Cant' open file: $!\n";
while (<TABLE>) {
        ($vlan, $unicast_mac, $mod_ports) = split (/,/, $_);
        write;
}
exit;

format STDOUT =
set cam permanent @<<<<<<<<<<<<<<<<< @<<< @<
$unicast_mac, $mod_ports, $vlan
```

Appendix A MIB Object Details

Object	tftpHost
OID	.1.3.6.1.4.1.9.5.1.5.1
Type	DisplayString
Permission	read-write
Syntax	OCTET STRING (0..64)
Status	Current
MIB	CISCO-STACK-MIB
Description	Name of source/destination host for the TFTP transfer or storage device transfer. If the name is for the TFTP transfer, it can be the IP address or

	the host name. If the name for the storage device transfer, it is in the format of deviceName: (e.g. slot0:, slot1:)
OID in Tree	::= { iso(1) org(3) dod(6) internet(1) private(4) enterprises(1) cisco(9) workgroup(5) ciscoStackMIB(1) tftpGrp(5) 1 }

Object	**tftpFile**
OID	**.1.3.6.1.4.1.9.5.1.5.2**
Type	DisplayString
Permission	read–write
Syntax	OCTET STRING (0..64)
Status	Current
MIB	CISCO–STACK–MIB
Description	Name of file for the TFTP transfer or for storage device transfer.
OID in Tree	::= { iso(1) org(3) dod(6) internet(1) private(4) enterprises(1) cisco(9) workgroup(5) ciscoStackMIB(1) tftpGrp(5) 2 }

Object	**tftpModule**
OID	**.1.3.6.1.4.1.9.5.1.5.3**
Type	Integer
Permission	read–write
Status	Current
Range	0 – 16
MIB	CISC O–STACK–MIB
Description	Which code/configuration of the module is transferred.
OID in Tree	::= { ISO(1) org(3) DOD(6) Internet(1) private(4) enterprises(1) cisco(9) workgroup(5) ciscoStackMIB(1) tftpGrp(5) 3 }

Object	**tftpAction**
OID	**.1.3.6.1.4.1.9.5.1.5.4**
Type	Integer
Permission	read–write
Status	Current
Values	1. other 2. downloadConfig 3. uploadConfig

	4. downloadSw
	5. uploadSw
	6. downloadFw
	7. uploadFw
MIB	CISCO–STACK–MIB
Description	If you set this object to one of the acceptable values, it initiates the requested action with the information given in tftpHost, tftpFile, tftpModule. downloadConfig(2): receive configuration from host/file uploadConfig(3): send configuration to host/file downloadSw(4): receive software image from host/file uploadSw(5): send software image to host/file downloadFw(6): receive firmware image from host/file uploadFw(7): send firmware image to host/file If you set this object to any other value, you get an error.
OID in Tree	::= { ISO(1) org(3) DOD(6) Internet(1) private(4) enterprises(1) cisco(9) workgroup(5) ciscoStackMIB(1) tftpGrp(5) 4 }

Object	**tftpResult**
OID	**.1.3.6.1.4.1.9.5.1.5.5**
Type	Integer
Permission	read–only
Status	Current
Values	1. inProgress
	2. success
	3. noResponse
	4. tooManyRetries
	5. noBuffers
	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

Description	Contains result of the last TFTP action request
OID in Tree	::= { ISO(1) org(3) DOD(6) Internet(1) private(4) enterprises(1) cisco(9) workgroup(5) ciscoStackMIB(1) tftpGrp(5) 5 }

Related Information

- **Moving Files and Images Between a Router and TFTP Server via SNMP**
- **Cisco MIBs Download**
- **Technical Support & Documentation – Cisco Systems**