Autoprovisioning Data Center Top-of-Rack Switches

What You Will Learn

With the vast increase in available information, movement to virtualization, and growing demand for cloud computing, data centers are expanding at a rapid pace. To save costs, increase availability, and address environmental factors, companies are expanding their data centers into different geographical areas, while their network operations groups are mainly staying in headquarters. This divergence creates a challenge for the network operations group, which must determine how to efficiently and quickly provision new deployments and configure replacements. The Cisco Catalyst® 4900 Series Switches, which are data center top-of-rack (ToR) switches running feature-rich Cisco IOS® Software, offer a solution. By using the AutoInstall and Cisco IOS Embedded Event Manager (EEM) features within Cisco IOS Software, the process of provisioning the switch can be automated.

This solution enables the facilities team to build server racks with ToR switches before moving the racks to their final location, thus providing a flexible and modular design. After the rack is cabled and powered on in the data center, the ToR switch autoprovisions itself.

Before looking at the details of this solution, you should examine the two features that are involved in this solution: AutoInstall and Cisco IOS EEM.

AutoInstall

AutoInstall allows network managers to connect a new Cisco® device to the network, power on the new device, and have the device configured automatically from a preexisting configuration file. The AutoInstall process begins when a device that uses Cisco IOS Software is turned on and a valid configuration file is not found in nonvolatile random-access memory (NVRAM).

1. On a switch when a LAN interface is used to connect to the network, AutoInstall will attempt to acquire an IP address for the attached interface using Dynamic Host Configuration Protocol (DHCP) requests, Bootstrap Protocol (BOOTP) requests, or Reverse Address Resolution Protocol (RARP) requests.
2. If a DHCP server responds, any or all of the following information can be returned to the new switch:
   - The IP address (yiaddr) and subnet mask (option 1) to be assigned to the interface on the new switch (the values in parentheses in this section represent the field names for the packets as defined in the relevant RFCs): The following lines will be written to the configuration of the new switch:

```
Interface <type><number>
   Ip address dhcp
```

   - The address of the Trivial FTP (TFTP) server (siaddr) to be used for AutoInstall requests.
   - The name of the configuration file (file or option 67) to be requested from the TFTP server.
   - The IP address of the TFTP server (option 150).
   - The hostname of the TFTP server (option 66 or sname): Typically either the TFTP address or name is specified, not both. If only the name of the TFTP server is specified, a Domain Name System (DNS) server must be available to translate the name to an IP address.
   - The IP address of up to two DNS servers (option 6): You should configure this option to be returned from the DHCP server only if the DNS server is in the same LAN as the interface performing the AutoInstall operation.
   - The IP address of the staging router (option 3): This option is provided for those cases in which the TFTP server is not on the same LAN segment as the new router, or in which the network topology requires the use

of a specific router. The staging router address is used to specify the router through which the AutoInstall TFTP requests should be sent (in other words, the first-hop router). This staging router is also referred to as the default or helper router. Only one staging router can be specified.

3. After the new switch acquires an IP address, it will automatically attempt to download a configuration file using one of the following methods:
   - If the host-specific configuration file (bootfile) name was specified by the DHCP server, that specific filename is used in the TFTP request. The new router will make three unicast TFTP requests for the specified bootfile. If the unicast attempts fail, or if a TFTP server address was not provided, the new router will make three broadcast requests to any available TFTP server for the specified bootfile.
   - If the specified bootfile cannot be located, or the new router was not provided a specific bootfile name by the DHCP server, AutoInstall will unicast or broadcast TFTP requests for a default network configuration file. The files requested are first for "network-confg" and then for "cisconet.cfg". The default network configuration file should have IP address to hostname mappings using ip host *ip address hostname* entries. If a command line entry for the IP address of the new router is not included in the configuration file, AutoInstall will attempt to resolve its hostname using a DNS query. If the new router can determine its hostname, a TFTP request will then be sent for the *"hostname-confg" or "hostname.cfg"* file. The *hostname* variable is replaced by the first eight characters of the new router's hostname. If the new router is unable to map its IP address to a hostname, AutoInstall will send TFTP requests for the default configuration file "router-confg" or "router.cfg."
4. The switch loads the configuration from the TFTP server.

For more information about AutoInstall, please see http://www.cisco.com/en/US/docs/ios/12_2/configfun/configuration/guide/fcf002.html.

Cisco IOS EEM

Cisco IOS EEM is a unique subsystem within Cisco IOS Software. Cisco IOS EEM is a powerful and flexible tool for automating tasks and customizing the behavior of Cisco IOS Software and device operation. Customers can use Cisco IOS EEM to create and run programs or scripts directly on a router or switch. The scripts are referred to as Cisco IOS EEM policies and can be programmed using a simple command-line interface (CLI) or a scripting language called Tool Command Language (TCL). Cisco IOS EEM allows customers to use the significant intelligence within Cisco IOS Software to respond to real-time events, automate tasks, create customer commands, and take local automated action based on conditions detected by the Cisco IOS Software itself.

For more information about Cisco IOS EEM, please see http://www.cisco.com/en/US/products/ps6815/products_ios_protocol_group_home.html.

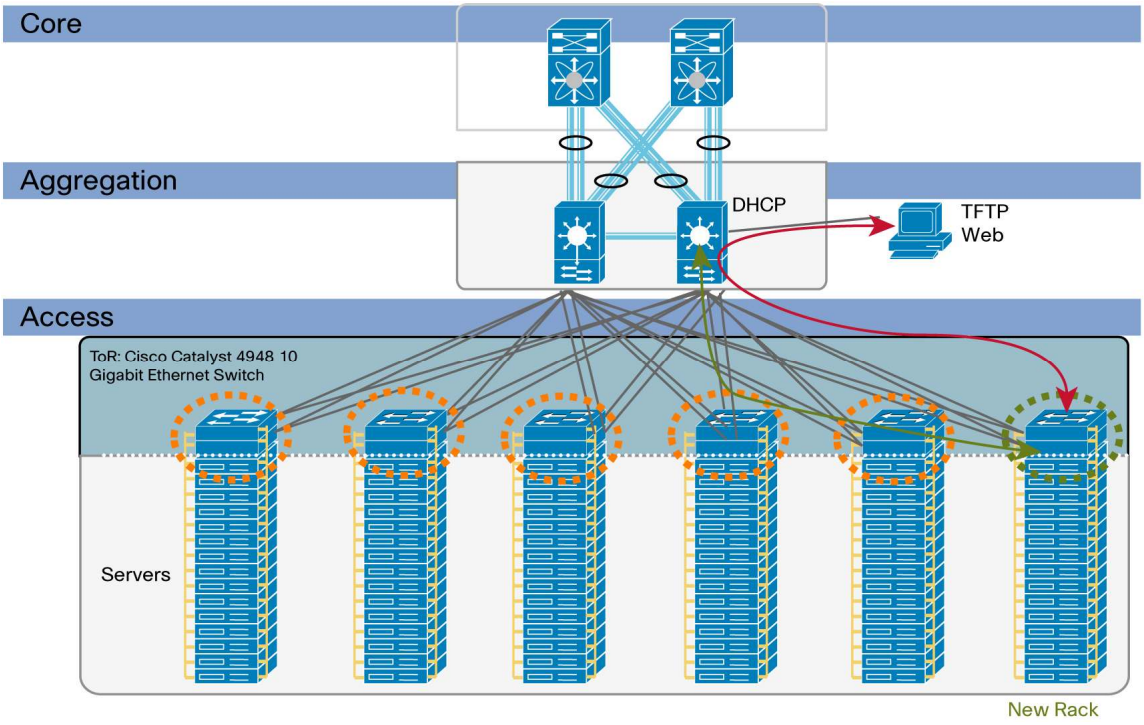
Autoprovisioning ToR Switches: Solution Overview

You now should understanding how AutoInstall works and the capabilities of Cisco IOS EEM, Now look at how you can use these two features to autoprovision the ToR device. A typical data center topology consists of core, aggregation, and access layers. The access layer here is the ToR switch, which is where the Cisco Catalyst 4900 Series is positioned. This solution uses the Cisco Catalyst 4948 10 Gigabit Ethernet Switch as the ToR switch, and its uplink is connected to the trunk port on an aggregation-layer switch, which here is the Cisco Catalyst 4900M Switch. Figure 1 provides a physical view of the setup. In this scenario, the facilities team has already racked and cabled the servers and the ToR switch. This rack is rolled to the production floor, and the power cables and uplink network cables are connected. Half an hour or so after the switch is powered on, it is provisioned and ready to go. That is what this solution does.

At a high level, this solution involves two stages:

1. The first stage uses AutoInstall. After the switch is powered on and booted, it acquires an IP address and downloads a common configuration file.
2. The second stage uses the capability of Cisco IOS EEM. This stage is highly customizable and can be changed to fit the customer's needs. In this example, Cisco IOS EEM scripts are programmed to upload some system information and download the necessary Cisco IOS Software image and location-specific configuration.

Figure 1. Physical Topology



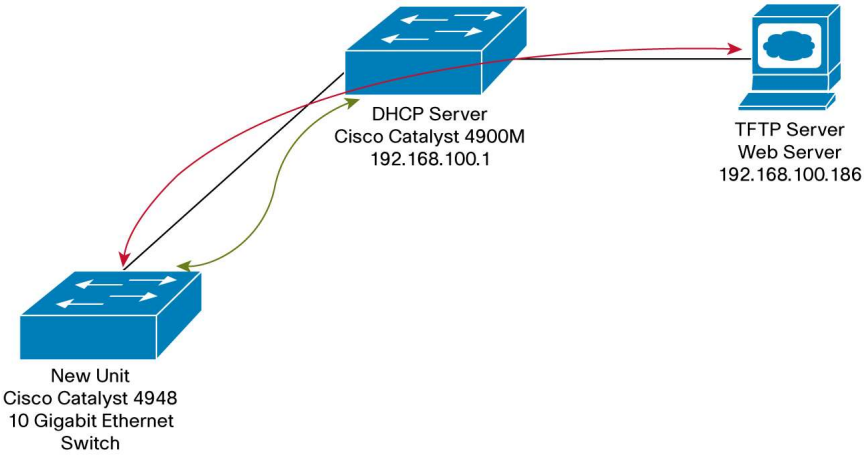
Now look at this solution in more detail. Any typical networking environment includes a DHCP server. In this example, the DHCP server is in the aggregation device (using the DHCP feature in Cisco IOS Software). This deployment includes different DHCP servers for different aggregation devices. With this information, you can use the IP address to identify the location of the newly installed rack: that is, the aggregation device to which this rack is connected.

The TFTP and web servers are on a centralized server, which can be in the same data center or outside the data center. In either case, all the server needs is reachability from the aggregation devices to the server.

For a deployment that uses a centralized DHCP server, the IP address will not reveal the location of the newly installed rack. In that case, a different method, such as the output of the Cisco Discovery Protocol (CDP) from the ToR switch, can be used to determine the location of the installed switch.

Figure 2 shows the logical components of this solution: the new ToR switch (Cisco Catalyst 4948 10 Gigabit Ethernet Switch), DHCP server (Cisco Catalyst 4900M), TFTP server, and web server. The TFTP and web servers are on the same physical server. The green line indicates communications between the ToR switch and the DHCP server, which consists mainly of DHCP messages. The red line indicates communications between the ToR switch and the TFTP and web servers. The communication on the red line is mainly TFTP and HTTP.

Figure 2. Logical Setup



Solution Requirements

The requirements for implementing this solution are as follows:

- Cisco Catalyst 4900 Series Switch that does not contain any valid configuration in NVRAM
- Cisco IOS Software Release 12.2(40)SG or later on the Cisco Catalyst 4900 Series
- DHCP server, TFTP server, and web server reachable from the aggregation switch

The first requirement (no valid configuration in NVRAM) helps ensure that AutoInstall will be triggered. This is the default behavior when the switch is shipped.

The second requirement indicates the minimum version of the software on the Cisco Catalyst 4900 Series platform that supports Cisco IOS EEM.

The DHCP server needs to support the DHCP options, mainly options 67 (configuration file name) and 150 (TFTP IP address). The TFTP server contains the necessary configuration files, Cisco IOS EEM TCL script, and Cisco IOS Software image. The web server hosts the CGI script that will interact with the switch.

The initial task is to create the following files and upload the required image on the TFTP server:

- Initial common configuration file that contains the Cisco IOS EEM applet on TFTP server (Appendix A: Initial Configuration")
- Perl script on a web server (Appendix B: CGI Script on the Web Server")
- Cisco IOS EEM TCL script on the TFTP server (Appendix C: Cisco IOS EEM TCL Script")
- Specific configuration file for the device (Appendix D: Device-Specific Configuration")

Initial Common Configuration File

The initial configuration file we use in this setup is c4948_init.cfg, which can be found in Appendix A. It basically consists of an EEM environment variable with the URL pointing to the location of the Perl script and a Cisco IOS EEM applet. The Cisco IOS EEM applet does the following:

- Triggers a countdown timer
- Copies the Cisco IOS EEM TCL script (autoconf.tcl) from the TFTP server
- Defines the Cisco IOS EEM policy directory
- Registers the TCL script

- Removes the Cisco IOS EEM applet (itself)

Note: This example also enables TCL debugging to give more detail to the reader. This is not needed in a production environment.

You may also want to include other elements in the initial configuration file. For example, if you are getting the TCl script from a TFTP server that is not yet reachable because there is no route to it, you can specify a static route for that script in the initial configuration file.

Perl Script on the Web Server

The Perl script is closely tied to the Cisco IOS EEM TCL script. In this setup, the Perl script is expecting the following information from the switch:

- The IP address the switch received from the DHCP server
- The version of the software that is running on the switch
- All the files that are currently in the switch's bootflash memory

The Perl script on the web server processes the information listed here and returns the following four pieces of information:

- Action: The action can be one of the following keywords:
  - None: No image download or reload is needed; the system is running the required image.
  - Load: Image is on the local file system. No need to copy, but need to reload the system with the required image.
  - Copy: Need to copy the new image to the local file system and reload the system with this new image.
- TFTP server IP address: This is the IP address of the TFTP server. This is where the required image and configuration file are stored.
- Name of the image: This is the name of the image that the switch should run.
- Configuration file: This is the name of the configuration file for this specific ToR switch based on the IP address that it received.

Cisco IOS EEM TCL Script

The Cisco EEM TCL script is one of the major components of this solution. This script is triggered by a countdown timer. After this timer is triggered, the script does the following:

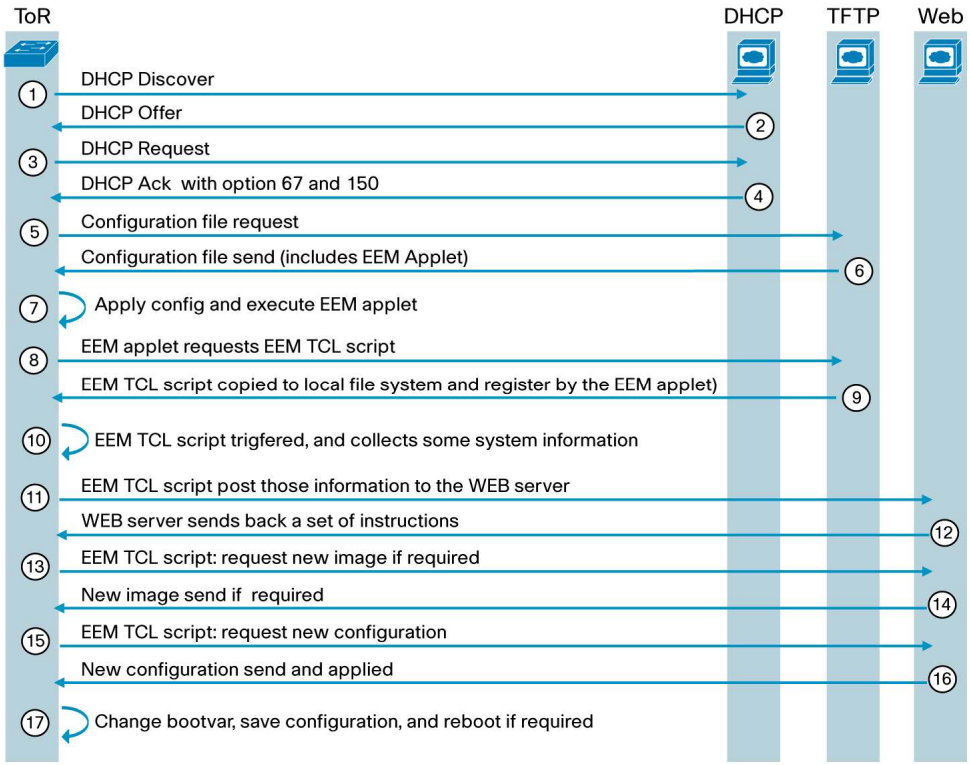
1. It runs CLI commands to collect:
   - IP address of the active interface
   - Running software version
   - Files stored on the local file system
2. It sends the collected information to the web server.
3. The web server's Perl script processes that information and sends back the action, TFTP server IP address, name of the image, and configuration file.
4. If the action is Copy, the script copies the required image from the TFTP server.
5. It copies the configuration file to local file system as well as to the running configuration.
6. It configures a Cisco IOS EEM applet to remove this Cisco IOS EEM TCL script (autoconfig.tcl).
7. If the action is Copy or Load, the script changes the configuration to boot from the required image.
8. If a reload is not required, the script generates a crypto key for Secure Shell (SSH).

9. If a reload is required, the script configures a Cisco IOS EEM applet (key_gen) to generate the crypto key after a reload.
10. It saves the running configuration to NVRAM.
11. It creates a temporary Cisco IOS EEM applet that is triggered on reload.
12. After reload is performed, the Cisco IOS EEM applet is triggered with a countdown timer, it generates a crypto key, and it removes itself.

Solution Process

Figure 3 shows the entire process from after the rack is installed in the data center and powered on to when it is provisioned with the specific configuration.

Figure 3. Autoprovision Process



Three different scenarios can occur after the new rack is powered on:

- The new switch needs to download the required image and reload with this image. The console log for this scenario is shown in Appendix E: Console Logs When the Script Needs to Download a New Image."
- The new switch is already running the required image and a reload is not needed. The console log for this scenario is shown in Appendix F: Console Logs When the System Is Running the Correct Image."
- The new switch is not running the required image, but the required image is already on the local file system, so a bootvar change and subsequent reload is needed. The console log for this scenario is shown in Appendix G: Console Logs When the Required Image Is on the Local File System, but the File System Is Not Running It."

As you can see from the console log, in all the preceding scenarios the switch is provisioned with the required image and configuration after this process completes. This is just one example of how the rich Cisco IOS Software features can help reduce operating costs. For more information about the Cisco Catalyst 4900 Series Switches, please refer to the "For More Information" section at the end of this document.

Appendix A: Initial Configuration

```
[root@gsbu-tme-186 tftpboot]# cat c4948_initial.cfg
event manager environment _autoconf_url http://192.168.100.186/cgi-bin/autoinstall/autoconf.cgi
event manager applet copy_script
 event timer countdown name 60second time 60 maxrun 900
 action 01.0 cli command "enable"
 action 02.0 cli command "config t"
 action 03.0 cli command "file prompt quiet"
 action 04.0 cli command "end"
 action 05.0 cli command "copy tftp://192.168.100.186/autoconf.tcl bootflash:"
 action 06.0 syslog msg "done with tc autoconf copy"
 action 07.0 cli command "config t"
 action 08.0 cli command "no file prompt quiet"
 action 09.0 cli command "event manager directory user policy bootflash:"
 action 0a.0 cli command "event manager policy autoconf.tcl"
 action 0b.0 syslog msg "done with registering autoconf tcl script"
 action 0c.0 cli command "no event manager applet copy_script"
 action 0d.0 cli command "end"
 action 0d.1 cli command "debug event manager tcl cli"
 action 0e.0 cli command "wr"
 action 0f.0 syslog msg "remove myself and save the configuration"
!
end
[root@gsbu-tme-186 tftpboot]#
```

Appendix B: CGI Script on the Web Server

```
[root@gsbu-tme-186 autoinstall]# cat autoconf.cgi
#!/usr/bin/perl
#--------------------------------------------------------------------------
#
# Name: autoconf.cgi
#
# This perl cgi script is called to send
# tftp server ip, config file, and image name information back to
# caller, given an ip address.
#
# Copyright (c) 2009 by cisco Systems, Inc.
# All rights reserved.
#--------------------------------------------------------------------------
use Getopt::Long;
$versionmatch = "12.2\\(50\\).*";
$serverback = "192.168.100.186";
$imageback = "cat4500-ipbasek9-mz.122-50.SG";
$imagebackpath = "/images";
$configback = "tor.cfg";
$configbackpath = "/images";
read(STDIN,$in,$ENV{'CONTENT_LENGTH'});
```

```
$incomingip = $in;
print "Content-type: text/html\n\n";
($version, $ip, $dir_info) = split('\n', $in, 3);
open(OUTFILE, "> outlog");

print OUTFILE "directory information:\n$dir_info\n";
# Find list of files on destination drive
$imagefoundondir = 0;
foreach $line (split('\n', $dir_info)) {
    ($j, $j, $j, $j, $j, $j, $j, $j, $j, $image) = split('\s+', $line);
    if ($image ne "") {
        if ($image eq $imageback) {
            $imagefoundondir = 1;
            print OUTFILE "matched image is $image\n";
            last;
        }
    }
}

print OUTFILE "IMAGEFOUND: $imagefoundondir\n";
print "server $serverback\n";
print OUTFILE "server $serverback\n";
if ($version =~ m/$versionmatch/) {
    print "action none\n";
    print "image null\n";
    print OUTFILE "action none\n";
    print OUTFILE "image null\n";
} else {
    if ($imagefoundondir == 1) {
        print "action load\n";
        print "image $imageback\n";
        print OUTFILE "action load\n";
        print OUTFILE "image $imageback\n";
    } else {
        $full_image_path = $imagebackpath . "/" . $imageback;
        print "action copy\n";
        print "image $full_image_path\n";
        print OUTFILE "action copy\n";
        print OUTFILE "image $full_image_path\n";
    }
}
$full_config_path = $configbackpath . "/" . $configback;
print "config $full_config_path";
print OUTFILE "config $full_config_path";
close OUTFILE;
 [root@gsbu-tme-186 autoinstall]#
```

Appendix C: Cisco IOS EEM TCL Script

```
[root@gsbu-tme-186 tftpboot]# cat autoconf.tcl
::cisco::eem::event_register_timer countdown name reloadtimer sec 15 maxrun 1900
```

```
#------------------------------------------------------------------
# EEM policy which triggers 15 seconds after register and sends
# ip info to the specified URL.
#
# May 2009, Cisco EEM team
#
# Copyright (c) 2009 by cisco Systems, Inc.
# All rights reserved.
#------------------------------------------------------------------
### The following EEM environment variables are used:
###
### _autoconf_url (mandatory)     - The URL location to which the
###                                 ip address is sent.
### Example: _autoconf_url          http://customerurlloc.com/autoconf.cgi
###
### _autoconf_dest                - The directory from which to search and
###                                 copy files to. Default is "bootflash:"
###
### _autoconf_debug (optional)    - Display policy debugging information.
### Example: _autoconf_debug        1
###
namespace import ::cisco::eem::*
namespace import ::cisco::lib::*
proc copy_file_from_tftp {server file destination} {
    global errorInfo
    global autoconf_debug
    if [catch {cli_open} result] {
        error $result $errorInfo
    } else {
        array set cli1 $result
    }
    if [catch {cli_exec $cli1(fd) "enable"} result] {
        error $result $errorInfo
    }
    if [catch {cli_exec $cli1(fd) "configure terminal"} result] {
        error $result $errorInfo
    }
    if [catch {cli_exec $cli1(fd) "file prompt quiet"} result] {
        error $result $errorInfo
    }
    if [catch {cli_exec $cli1(fd) "end"} result] {
        error $result $errorInfo
    }
    if [catch {cli_exec $cli1(fd) "copy tftp://$server/$file $destination"} \
        result] {
        error $result $errorInfo
    }
    if [catch {cli_exec $cli1(fd) "configure terminal"} result] {
        error $result $errorInfo
    }
```

```
    if [catch {cli_exec $cli1(fd) "no file prompt quiet"} result] {
        error $result $errorInfo
    }
    if [catch {cli_exec $cli1(fd) "end"} result] {
        error $result $errorInfo
    }
    if [catch {cli_close $cli1(fd) $cli1(tty_id)} result] {
        error $result $errorInfo
    }
    if {$autoconf_debug == 1} {
        puts "copied $server $file to $destination"
    }
}
proc copy_to_running {server file} {
    global errorInfo
    global autoconf_debug
    if [catch {cli_open} result] {
        error $result $errorInfo
    } else {
        array set cli1 $result
    }
    if [catch {cli_exec $cli1(fd) "enable"} result] {
        error $result $errorInfo
    }
    if [catch {cli_exec $cli1(fd) "configure terminal"} result] {
        error $result $errorInfo
    }
    if [catch {cli_exec $cli1(fd) "file prompt quiet"} result] {
        error $result $errorInfo
    }
    if [catch {cli_exec $cli1(fd) "end"} result] {
        error $result $errorInfo
    }
    if [catch {cli_write $cli1(fd) "copy tftp://$server/$file running"} \
        result] {
        error $result $errorInfo
    }
    if [catch {cli_read_pattern $cli1(fd) "(#|>)"} result] {
        error $result $errorInfo
    }
    if [catch {cli_exec $cli1(fd) "configure terminal"} result] {
        error $result $errorInfo
    }
    if [catch {cli_exec $cli1(fd) "no file prompt quiet"} result] {
        error $result $errorInfo
    }
    if [catch {cli_exec $cli1(fd) "end"} result] {
        error $result $errorInfo
    }
    if [catch {cli_close $cli1(fd) $cli1(tty_id)} result] {
```

```
        error $result $errorInfo
    }
    if {$autoconf_debug == 1} {
        puts "copied $server $file to $destination"
    }
}
if {[info exists _autoconf_debug]} {
    set autoconf_debug 1
} else {
    set autoconf_debug 0
}
if {![info exists _autoconf_dest]} {
    set _autoconf_dest "bootflash:"
}
# 2. Gather interface and version information from CLI
if [catch {cli_open} result] {
    error $result $errorInfo
} else {
    array set cli1 $result
}
if [catch {cli_exec $cli1(fd) "enable"} result] {
    error $result $errorInfo
}
if [catch {cli_exec $cli1(fd) \
      "show ip int brief | include up                    up"} cli_show_up] {
    error $cli_show_up $errorInfo
}
if [catch {cli_exec $cli1(fd) \
      "show version | include Cisco IOS Software"} cli_version] {
    error $cli_version $errorInfo
}
if [catch {cli_exec $cli1(fd) "dir $_autoconf_dest"} dir_info] {
    error $dir_info $errorInfo
}
if [catch {cli_close $cli1(fd) $cli1(tty_id)} result] {
    error $result $errorInfo
}
#Check if _autoconf_dest exists
set i [regexp {%Error opening.*} $dir_info match]
if {$i == 1} {
    action_syslog priority info msg \
                "Unable to find directory: $_autoconf_dest, exiting"
    return;
}
#Get first assigned interface
set match_found 0
foreach cli_line [split $cli_show_up \n] {
    set i [regexp {(\S+)\s+(\S+)\s+YES.*} $cli_line match interface ip]
    if {$i == 1} {
        if {[string equal $ip "unassigned"]} {
```

```
            continue
        }
        set match_found 1
        break
    }
}
if {$match_found == 1} {
    if {$autoconf_debug == 1} {
        action_syslog priority info msg "Found:  --$interface-- --$ip--"
    }
} else {
    action_syslog priority info msg \
                "Unable to match assigned interface, exiting"
    return;
}
# Process version
set version_match [regexp {.* Version (\S+).*} $cli_version match mversion]
if {$version_match == 1} {
    if {$autoconf_debug == 1} {
        action_syslog priority info msg "Version Match Found:  $mversion"
    }
} else {
    set mversion "unknown"
    action_syslog priority info msg "Unable to find software version"
}
set senddata "$mversion\n"
append senddata "$ip\n"
append senddata "$dir_info"
#Determine URL location to send info to.
if {[info exists _autoconf_url]} {
    set postUrl $_autoconf_url
} else {
    action_syslog priority info msg \
        "EEM environment variable _autoconf_url not set, exiting"
    return
}
if {$autoconf_debug == 1} {
    action_syslog priority info msg "Using autoconf URL: $postUrl"
}
for {set i 0} {($i < 25) && \
    [catch {http::geturl $postUrl -query $senddata -queryblocksize 50 -type \
            "text/plain" } token]} {incr i} {
    after 2000
}
if {$autoconf_debug == 1} {
    action_syslog priority info msg \
        "Response from URL: [http::data $token]"
}
# 6. if the transmission to the autoconf site was successful,
#    log a message
```

```
if {$i < 5} {
    if {[http::error $token] == ""} {
        foreach line [split [http::data $token] \n] {
            set field [lindex [split $line] 0]
            set value [lindex [split $line] 1]
            puts "field: $field  value: $value"
            switch $field {
action { set action $value }
                server { set server $value }
                image { set image $value }
                config { set config $value }
            }
        }
        # Process server result here
        if {$server != ""} {
    if {$action == "copy"} {
if {$image == "null"} {
                    puts "Error: action should not be copy without image info."
                    return;
                }
                puts "start copying image $image from $server"
                copy_file_from_tftp $server $image $_autoconf_dest
                puts "done copying image"
                set reload_needed 1
            } elseif {$action == "load"} {
puts "Image exists on $_autoconf_dest, reload without copying"
                set reload_needed 1
    } else {
                puts "No need to copy the image"
                set reload_needed 0
            }
            puts "start copying configuration file"
            copy_file_from_tftp $server $config $_autoconf_dest
            copy_to_running  $server $config
            puts "done copying configuration file"
        } else {
            puts "Error: No copy information received from http server."
        }
    }
    http::Finish $token
    http::cleanup $token
} else {
    action_syslog priority notice msg \
        "autoconf delivery failed"
    http::Finish $token
    http::cleanup $token
}
#Have policy remove itself
if [catch {cli_open} result] {
    error $result $errorInfo
```

```
} else {
    array set cli1 $result
}
if [catch {cli_exec $cli1(fd) "enable"} result] {
    error $result $errorInfo
}
if [catch {cli_exec $cli1(fd) "configure terminal"} result] {
    error $result $errorInfo
}
if [catch {cli_exec $cli1(fd) "no event manager policy autoconf.tcl"} result] {
    error $result $errorInfo
}
if {$reload_needed == 1} {
    if [catch {cli_exec $cli1(fd) "no boot system"} result] {
        error $result $errorInfo
    }
    regsub "^.*/" $image "" name
    if [catch {cli_exec $cli1(fd) "boot system bootflash:$name"} result] {
        error $result $errorInfo
    }
    if [catch {cli_exec $cli1(fd) "config-register 0x2102"} result] {
        error $result $errorInfo
    }
    puts "done with changing the boot parameters"
}

if [catch {cli_exec $cli1(fd) "end"} result] {
    error $result $errorInfo
}
if {$reload_needed != 1} {
  # Reload is not needed, generate crypto key locally.
  if [catch {cli_exec $cli1(fd) "config t"} result] {
       error $result $errorInfo
  }
  if [catch {cli_write $cli1(fd) "crypto key zeroize rsa"} result] {
       error $result $errorInfo
  }

  if [catch {cli_read_pattern $cli1(fd) "no.*:|in configuration"} result] {
      error $result $errorInfo
  }

  if {[string first "no" $result] != -1} {
      if [catch {cli_exec $cli1(fd) "yes"} result] {
         error $result $errorInfo
      }
  }
  # Temp set domain to cisco.com
  if [catch {cli_exec $cli1(fd) "ip domain-name cisco.com"} result] {
      error $result $errorInfo
```

```
  }
  if [catch {cli_write $cli1(fd) "crypto key generate rsa"} result] {
      error $result $errorInfo
  }
  if [catch {cli_read_pattern $cli1(fd) "How many bits in the modulus .*:"} result]
{
      error $result $errorInfo
  }
  if [catch {cli_exec $cli1(fd) "1024"} result] {
      error $result $errorInfo
  }
  if [catch {cli_exec $cli1(fd) "end"} result] {
      error $result $errorInfo
  }

  action_syslog priority info msg "Crypto key generation has completed."
} else {
    # Reload is needed, create applet to handle key generation post reload
    if [catch {cli_exec $cli1(fd) "config t"} result] {
        error $result $errorInfo
    }
    # Temp set domain to cisco.com
    if [catch {cli_exec $cli1(fd) "ip domain-name cisco.com"} result] {
        error $result $errorInfo
    }
    if [catch {cli_exec $cli1(fd) "ev man app key_gen"} result] {
        error $result $errorInfo
    }
    if [catch {cli_exec $cli1(fd) "ev timer countdown time 120 name key maxrun
300"} result] {
        error $result $errorInfo
    }
   if [catch {cli_exec $cli1(fd) "action 1 cli command \"enable\""} result] {
       error $result $errorInfo
   }
   if [catch {cli_exec $cli1(fd) "action 2 cli command \"config t\""} result] {
       error $result $errorInfo
   }
   if [catch {cli_exec $cli1(fd) "action 3 cli command \"crypto key generate rsa\"
pattern \"How many bits in the modulus\""} result] {
       error $result $errorInfo
   }
   if [catch {cli_exec $cli1(fd) "action 4 cli command \"1024\""} result] {
       error $result $errorInfo
   }

   if [catch {cli_exec $cli1(fd) "action 5 syslog msg \"Crypto key generation has
completed.\""} result] {
       error $result $errorInfo
   }
```

```
   if [catch {cli_exec $cli1(fd) "action 6 cli command \"no ev man app key_gen\""}
result] {
       error $result $errorInfo
   }
   if [catch {cli_exec $cli1(fd) "end"} result] {
       error $result $errorInfo
   }
}
after 20000
if [catch {cli_write $cli1(fd) "copy run start"} result] {
    error $result $errorInfo
}
if [catch {cli_read_pattern $cli1(fd) "\\?"} result] {
    error $result $errorInfo
}
if [catch {cli_exec $cli1(fd) "\n"} result] {
    error $result $errorInfo
}
# The following configurations will never be saved.
if [catch {cli_exec $cli1(fd) "configure terminal"} result] {
    error $result $errorInfo
}
if [catch {cli_exec $cli1(fd) "ev man app reload_helper"} result] {
    error $result $errorInfo
}
if [catch {cli_exec $cli1(fd) "ev app sub 798 type 1"} result] {
    error $result $errorInfo
}
if [catch {cli_exec $cli1(fd) "action 1 reload"} result] {
    error $result $errorInfo
}
if [catch {cli_exec $cli1(fd) "end"} result] {
    error $result $errorInfo
}
if [catch {cli_close $cli1(fd) $cli1(tty_id)} result] {
    error $result $errorInfo
}
## reload if needed
if {$reload_needed == 1} {
    after 10000
    # action_reload
    event_publish sub_system 798 type 1 arg1 none
}
 [root@gsbu-tme-186 tftpboot]#
```

Appendix D: Device-Specific Configuration

```
[root@gsbu-tme-186 images]# cat tor.cfg
!
service timestamps debug datetime localtime show-timezone
service timestamps log datetime localtime show-timezone
```

```
service password-encryption
service compress-config
!
clock timezone PST -8
!
!
hostname C4948-36
!
boot-start-marker
boot-end-marker
!
!
system mtu 1552
power redundancy-mode redundant
!
spanning-tree mode pvst
spanning-tree extend system-id
no spanning-tree vlan 1-4094
!
clock summer-time PDT recurring
!
snmp-server user labRW labRW v1 access 2
snmp-server user labRW labRW v2c access 2
snmp-server community labRO RO 2
snmp-server community labRW RW 2
snmp-server location SJC-North
!
!
line con 0
 stopbits 1
line vty 0 4
 transport preferred none
 transport input ssh
line vty 5 15
 transport preferred none
 transport input ssh
!
end
 [root@gsbu-tme-186 images]#
```

Appendix E: Console Logs When the Script Needs to Download a New Image

```
switch#sho bootvar
BOOT variable =
CONFIG_FILE variable =
BOOTLDR variable does not exist
Configuration register is 0x2101
switch#dir bootflash:
Directory of bootflash:/
    1  -rwx    15155232  Aug 31 2009 11:07:48 -07:00  cat4500-ipbasek9-mz.122-46.SG
61341696 bytes total (46186336 bytes free)
```

```
switch#reload
Proceed with reload? [confirm]
*Mar 29 20:11:03 PDT: %SYS-5-RELOAD: Reload requested by console. Reload reason:
Reload command
************************************************************
*                                                          *
* Welcome to Rom Monitor for WS-C4948-10GE System.         *
* Copyright (c) 2003-2006 by Cisco Systems, Inc.           *
* All rights reserved.                                     *
*                                                          *
************************************************************
 Rom Monitor Program Version 12.2(31r)SGA1
 Supervisor: WS-C4948-10GE  Chassis: WS-C4948
 Hardware Revisions - Board: 8.5 CPLD Gill: 20
MAC Address  : 00-1f-9e-af-cc-3f
 Ip Address   : Not set.
 Netmask      : Not set.
 Gateway      : Not set.
 TftpServer   : Not set.
***** The system will autoboot in 5 seconds *****
Type control-C to prevent autobooting.
 . . . . .
 Established physical link 1Gb Full Duplex
 Network layer connectivity may take a few seconds
******** The system will autoboot now ********
config-register = 0x2101
Autobooting using the first file from bootflash.....
Rommon reg: 0x00004180
#############
k2diags version 5.2_c
prod: WS-C4948-10GE  part: 73-9344-10  serial: FOX1203G1XG
Power-on-self-test for Module 1:  WS-C4948-10GE
 Port/Test Status: (. = Pass, F = Fail, U = Untested)
Cpu Subsystem Tests ...
seeprom: . temperature_sensor: .
Port Traffic: L3 Serdes Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .
62: . 63: .
Port Traffic: L2 Serdes Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .
62: . 63: .
Port Traffic: L2 Asic Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
```

```
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .
62: . 63: .
Switch Subsystem Memory ...
 1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: . 12: .
13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: . 24: .
25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: . 36: .
37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: . 48: .
49: . 50: . 51: .
Module 1 Passed
Exiting to ios...
Rommon reg: 0x00000180
########################################
             Restricted Rights Legend
Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.
           cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706
Cisco IOS Software, Catalyst 4500 L3 Switch Software (cat4500-IPBASEK9-M), Version
12.2(46)SG, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2008 by Cisco Systems, Inc.
Compiled Fri 27-Jun-08 16:56 by prod_rel_team
Image text-base: 0x10000000, data-base: 0x11ABEC24
This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.
A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html
If you require further assistance please contact us by sending email to
export@cisco.com.
cisco WS-C4948-10GE (MPC8540) processor (revision 5) with 262144K bytes of memory.
Processor board ID FOX1203G1XG
MPC8540 CPU at 667Mhz, Fixed Module
Last reset from Reload
1 Virtual Ethernet interface
48 Gigabit Ethernet interfaces
2 Ten Gigabit Ethernet interfaces
511K bytes of non-volatile configuration memory.
Press RETURN to get started!
```

```
*Mar  1 00:00:01.323: %C4K_IOSSYS-3-BLANKSTARTUPCONFIG: Blank or invalid startup-
config, booting up with defaults

*Mar 30 03:12:12.527: %SPANTREE-5-EXTENDED_SYSID: Extended SysId enabled for type
vlan

*Mar 30 03:12:14.443: %C4K_IOSMODPORTMAN-6-MODULEONLINE: Module 1 (WS-C4948-10GE
S/N: FOX1203G1XG Hw: 3.4) is online

*Mar 30 03:12:33.575: %SYS-5-RESTART: System restarted --

Cisco IOS Software, Catalyst 4500 L3 Switch Software (cat4500-IPBASEK9-M), Version
12.2(46)SG, RELEASE SOFTWARE (fc1)

Technical Support: http://www.cisco.com/techsupport

Copyright (c) 1986-2008 by Cisco Systems, Inc.

Compiled Fri 27-Jun-08 16:56 by prod_rel_team

*Mar 30 03:12:35.759: %SPANTREE-7-RECV_1Q_NON_TRUNK: Received 802.1Q BPDU on non
trunk TenGigabitEthernet1/49 VLAN1.

*Mar 30 03:12:35.759: %SPANTREE-7-BLOCK_PORT_TYPE: Blocking TenGigabitEthernet1/49
on VLAN0001. Inconsistent port type.

*Mar 30 03:13:35.407: AUTOINSTALL: Vlan1 is assigned 192.168.100.231

*Mar 30 03:13:45.407: AUTOINSTALL: Obtain tftp server address (opt 150)
192.168.100.186

Loading c4948_initial.cfg from 192.168.100.186 (via Vlan1): !

[OK - 1004 bytes]

*Mar 30 03:13:58.459: %SYS-5-CONFIG_I: Configured from
tftp://192.168.100.186/c4948_initial.cfg by console

*Mar 30 03:14:58.499: %SYS-5-CONFIG_I: Configured from console by  on vty0
(EEM:copy_script)

*Mar 30 03:14:59.755: %HA_EM-6-LOG: copy_script: done with tc autoconf copy

*Mar 30 03:14:59.903: %HA_EM-6-LOG: copy_script: done with registering autoconf tcl
script

*Mar 30 03:14:59.915: %SYS-5-CONFIG_I: Configured from console by  on vty0
(EEM:copy_script)

*Mar 30 03:15:00.459: %HA_EM-6-LOG: copy_script: remove myself and save the
configuration

*Mar 30 03:15:15.047: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_open
called.

*Mar 30 03:15:15.179: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch>

*Mar 30 03:15:15.179: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch>enable

*Mar 30 03:15:15.291: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#

*Mar 30 03:15:15.291: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#show ip int brief | include up                      up

*Mar 30 03:15:15.403: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Vlan1
192.168.100.231 YES DHCP   up                    up

*Mar 30 03:15:15.403: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
TenGigabitEthernet1/49 unassigned      YES unset  up                      up

*Mar 30 03:15:15.403: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#

*Mar 30 03:15:15.403: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#show version | include Cisco IOS Software

*Mar 30 03:15:15.515: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Cisco IOS
Software, Catalyst 4500 L3 Switch Software (cat4500-IPBASEK9-M), Version
12.2(46)SG, RELEASE SOFTWARE (fc1)

*Mar 30 03:15:15.515: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#

*Mar 30 03:15:15.515: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#dir bootflash:

*Mar 30 03:15:15.627: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Directory
of bootflash:/

*Mar 30 03:15:15.627: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
```

```
*Mar 30 03:15:15.627: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :     1  -
rwx    15155232  Aug 31 2009 18:07:48 +00:00  cat4500-ipbasek9-mz.122-46.SG

*Mar 30 03:15:15.627: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :     2  -
rw-       12624  Mar 30 2010 03:14:59 +00:00  autoconf.tcl

*Mar 30 03:15:15.627: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :

*Mar 30 03:15:15.627: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : 61341696
bytes total (46173584 bytes free)

*Mar 30 03:15:15.627: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#

*Mar 30 03:15:15.627: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_close
called.

*Mar 30 03:15:21.003: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_open
called.

*Mar 30 03:15:21.119: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch>

*Mar 30 03:15:21.119: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch>enable

*Mar 30 03:15:21.231: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#

*Mar 30 03:15:21.231: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#configure terminal

*Mar 30 03:15:21.359: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.

*Mar 30 03:15:21.359: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#

*Mar 30 03:15:21.359: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#file prompt quiet

*Mar 30 03:15:21.571: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#

*Mar 30 03:15:21.571: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#end

*Mar 30 03:15:21.571: %SYS-5-CONFIG_I: Configured from console by  on vty0
(EEM:autoconf.tcl)

*Mar 30 03:15:21.683: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#

*Mar 30 03:15:21.683: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#copy tftp://192.168.100.186//images/cat4500-ipbasek9-mz.122-50.SG bootflash:

*Mar 30 03:17:53.863: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Accessing
tftp://192.168.100.186//images/cat4500-ipbasek9-mz.122-50.SG...

*Mar 30 03:17:53.863: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Loading
/images/cat4500-ipbasek9-mz.122-50.SG from 192.168.100.186 (via Vlan1):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

*Mar 30 03:17:53.863: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : [OK -
15728024 bytes]

*Mar 30 03:17:53.863: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :

*Mar 30 03:17:53.863: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : 15728024
bytes copied in 151.024 secs (104143 bytes/sec)

*Mar 30 03:17:53.863: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#

*Mar 30 03:17:53.863: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#configure terminal

*Mar 30 03:17:53.975: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.

*Mar 30 03:17:53.975: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#

*Mar 30 03:17:53.975: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#no file prompt quiet

*Mar 30 03:17:54.091: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#

*Mar 30 03:17:54.095: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#end
```

```
*Mar 30 03:17:54.095: %SYS-5-CONFIG_I: Configured from console by  on vty0
(EEM:autoconf.tcl)
*Mar 30 03:17:54.207: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#
*Mar 30 03:17:54.207: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_close
called.
*Mar 30 03:17:54.207: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_open
called.
*Mar 30 03:17:54.319: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch>
*Mar 30 03:17:54.319: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch>enable
*Mar 30 03:17:54.431: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#
*Mar 30 03:17:54.431: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#configure terminal
*Mar 30 03:17:54.543: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.
*Mar 30 03:17:54.543: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#
*Mar 30 03:17:54.543: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#file prompt quiet
*Mar 30 03:17:54.655: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#
*Mar 30 03:17:54.655: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#end
*Mar 30 03:17:54.655: %SYS-5-CONFIG_I: Configured from console by  on vty1
(EEM:autoconf.tcl)
*Mar 30 03:17:54.767: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#
*Mar 30 03:17:54.767: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#copy tftp://192.168.100.186//images/tor.cfg bootflash:
*Mar 30 03:17:54.979: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Accessing
tftp://192.168.100.186//images/tor.cfg...
*Mar 30 03:17:54.979: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Loading
/images/tor.cfg from 192.168.100.186 (via Vlan1): !
*Mar 30 03:17:54.979: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : [OK - 761
bytes]
*Mar 30 03:17:54.979: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
*Mar 30 03:17:54.979: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : 761 bytes
copied in 0.024 secs (31708 bytes/sec)
*Mar 30 03:17:54.979: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#
*Mar 30 03:17:54.979: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#configure terminal
*Mar 30 03:17:55.091: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.
*Mar 30 03:17:55.091: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#
*Mar 30 03:17:55.091: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#no file prompt quiet
*Mar 30 03:17:55.203: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#
*Mar 30 03:17:55.203: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#end
*Mar 30 03:17:55.203: %SYS-5-CONFIG_I: Configured from console by  on vty1
(EEM:autoconf.tcl)
*Mar 30 03:17:55.315: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#
*Mar 30 03:17:55.315: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_close
called.
*Mar 30 03:17:55.315: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_open
called.
```

```
*Mar 30 03:17:55.427: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch>
*Mar 30 03:17:55.427: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch>enable
*Mar 30 03:17:55.539: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#
*Mar 30 03:17:55.539: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#configure terminal
*Mar 30 03:17:55.651: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.
*Mar 30 03:17:55.651: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#
*Mar 30 03:17:55.651: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#file prompt quiet
*Mar 30 03:17:55.763: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#
*Mar 30 03:17:55.763: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#end
*Mar 30 03:17:55.763: %SYS-5-CONFIG_I: Configured from console by  on vty0
(EEM:autoconf.tcl)
*Mar 30 03:17:55.875: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#
*Mar 30 03:17:55.875: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#copy tftp://192.168.100.186//images/tor.cfg running
*Mar 29 19:18:04 PST: %SYS-6-CLOCKUPDATE: System clock has been updated from
03:18:04 UTC Tue Mar 30 2010 to 19:18:04 PST Mon Mar 29 2010, configured from
console by  on vty0 (EEM:autoconf.tcl).
*Mar 29 20:18:04 PDT: %SYS-6-CLOCKUPDATE: System clock has been updated from
19:18:04 PST Mon Mar 29 2010 to 20:18:04 PDT Mon Mar 29 2010, configured from
console by  on vty0 (EEM:autoconf.tcl).
*Mar 29 20:18:04 PDT: %SYS-5-CONFIG_I: Configured from
tftp://192.168.100.186//images/tor.cfg by  on vty0 (EEM:autoconf.tcl)
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Accessing
tftp://192.168.100.186//images/tor.cfg...
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Loading
/images/tor.cfg from 192.168.100.186 (via Vlan1): !
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : [OK - 761
bytes]
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Global
Ethernet MTU is set to 1552 bytes.
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Note:
this is the Ethernet payload size, not the total
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Ethernet
frame size, which includes the Ethernet
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
header/trailer and possibly other tags, such as ISL or
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : 802.1q
tags.
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : 761 bytes
copied in 9.104 secs (84 bytes/sec)
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
#configure terminal
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#no file prompt quiet
```

```
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#end
*Mar 29 20:18:05 PDT: %SYS-5-CONFIG_I: Configured from console by  on vty0
(EEM:autoconf.tcl)
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_close
called.
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_open
called.
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36>
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36>enable
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36#configure terminal
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#no event manager policy autoconf.tcl
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Mar 29 20:18:05 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#no boot system
*Mar 29 20:18:06 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Mar 29 20:18:06 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#boot system bootflash:cat4500-ipbasek9-mz.122-50.SG
*Mar 29 20:18:06 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Mar 29 20:18:06 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#config-register 0x2102
*Mar 29 20:18:06 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Mar 29 20:18:06 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#end
*Mar 29 20:18:06 PDT: %SYS-5-CONFIG_I: Configured from console by  on vty1
(EEM:autoconf.tcl)
*Mar 29 20:18:06 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Mar 29 20:18:06 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36#config t
*Mar 29 20:18:06 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.
*Mar 29 20:18:06 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Mar 29 20:18:06 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#ip domain-name cisco.com
*Mar 29 20:18:06 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Mar 29 20:18:06 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#ev man app key_gen
*Mar 29 20:18:06 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Mar 29 20:18:06 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#ev timer countdown time 120 name key maxrun 300
```

```
*Mar 29 20:18:06 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Mar 29 20:18:06 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#action 1 cli command "enable"
*Mar 29 20:18:06 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Mar 29 20:18:06 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#action 2 cli command "config t"
*Mar 29 20:18:07 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Mar 29 20:18:07 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#action 3 cli command "crypto key generate rsa" pattern "How many
bits in the modulus"
*Mar 29 20:18:07 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Mar 29 20:18:07 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#action 4 cli command "1024"
*Mar 29 20:18:07 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Mar 29 20:18:07 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#action 5 syslog msg "Crypto key generation has completed."
*Mar 29 20:18:07 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Mar 29 20:18:07 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#action 6 cli command "no ev man app key_gen"
*Mar 29 20:18:07 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Mar 29 20:18:07 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#end
*Mar 29 20:18:07 PDT: %SYS-5-CONFIG_I: Configured from console by  on vty1
(EEM:autoconf.tcl)
*Mar 29 20:18:07 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Mar 29 20:18:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36#copy run start
*Mar 29 20:18:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Destination filename [startup-config]?
*Mar 29 20:18:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
*Mar 29 20:18:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Building
configuration...
*Mar 29 20:18:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Compressed configuration from 3333 bytes to 1582 bytes[OK]
*Mar 29 20:18:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Mar 29 20:18:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Mar 29 20:18:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36#configure terminal
*Mar 29 20:18:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.
*Mar 29 20:18:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Mar 29 20:18:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#ev man app reload_helper
*Mar 29 20:18:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Mar 29 20:18:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#ev app sub 798 type 1
*Mar 29 20:18:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
```

```
*Mar 29 20:18:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#action 1 reload
*Mar 29 20:18:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Mar 29 20:18:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#end
*Mar 29 20:18:28 PDT: %SYS-5-CONFIG_I: Configured from console by  on vty1
(EEM:autoconf.tcl)
*Mar 29 20:18:29 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Mar 29 20:18:29 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_close
called.
*Mar 29 20:18:39 PDT: %SYS-5-RELOAD: Reload requested by EEM. Reload reason:
Embedded Event Manager action
**********************************************************
*                                                        *
* Welcome to Rom Monitor for WS-C4948-10GE System.       *
* Copyright (c) 2003-2006 by Cisco Systems, Inc.         *
* All rights reserved.                                   *
*                                                        *
**********************************************************
Rom Monitor Program Version 12.2(31r)SGA1
 Supervisor: WS-C4948-10GE  Chassis: WS-C4948
 Hardware Revisions - Board: 8.5 CPLD Gill: 20
MAC Address  : 00-1f-9e-af-cc-3f
 Ip Address   : Not set.
 Netmask      : Not set.
 Gateway      : Not set.
 TftpServer   : Not set.
***** The system will autoboot in 5 seconds *****
Type control-C to prevent autobooting.
 . . . . .
 Established physical link 1Gb Full Duplex
 Network layer connectivity may take a few seconds
******** The system will autoboot now ********
config-register = 0x2102
 Autobooting using BOOT variable specified file.....
Current BOOT file is --- bootflash:cat4500-ipbasek9-mz.122-50.SG
Rommon reg: 0x00004180
#############
k2diags version 5.2_d
prod: WS-C4948-10GE  part: 73-9344-10  serial: FOX1203G1XG
Power-on-self-test for Module 1:  WS-C4948-10GE
 Port/Test Status: (. = Pass, F = Fail, U = Untested)
Cpu Subsystem Tests ...
seeprom: . temperature_sensor: .
Port Traffic: L3 Serdes Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .
62: . 63: .
```

```
Port Traffic: L2 Serdes Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .
62: . 63: .
Port Traffic: L2 Asic Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .
62: . 63: .
Switch Subsystem Memory ...
 1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: . 12: .
13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: . 24: .
25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: . 36: .
37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: . 48: .
49: . 50: . 51: .
Module 1 Passed
Exiting to ios...
Rommon reg: 0x00000180
##########################################
             Restricted Rights Legend
Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.
           cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706
Cisco IOS Software, Catalyst 4500 L3 Switch Software (cat4500-IPBASEK9-M), Version
12.2(50)SG, RELEASE SOFTWARE (fc4)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2008 by Cisco Systems, Inc.
Compiled Wed 24-Dec-08 18:23 by prod_rel_team
Image text-base: 0x10000000, data-base: 0x11C2928C
This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.
A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html
If you require further assistance please contact us by sending email to
export@cisco.com.
```

```
cisco WS-C4948-10GE (MPC8540) processor (revision 5) with 262144K bytes of memory.
Processor board ID FOX1203G1XG
MPC8540 CPU at 667Mhz, Fixed Module
Last reset from Reload
1 Virtual Ethernet interface
48 Gigabit Ethernet interfaces
2 Ten Gigabit Ethernet interfaces
511K bytes of non-volatile configuration memory.
Press RETURN to get started!
*Mar 30 03:19:12.539: %SPANTREE-5-EXTENDED_SYSID: Extended SysId enabled for type
vlan
*Mar 29 19:19:14 PST: %SYS-6-CLOCKUPDATE: System clock has been updated from
03:19:14 UTC Tue Mar 30 2010 to 19:19:14 PST Mon Mar 29 2010, configured from
console by console.
*Mar 29 20:19:14 PDT: %SYS-6-CLOCKUPDATE: System clock has been updated from
19:19:14 PST Mon Mar 29 2010 to 20:19:14 PDT Mon Mar 29 2010, configured from
console by console.
*Mar 29 20:19:14 PDT: %C4K_IOSMODPORTMAN-6-MODULEONLINE: Module 1 (WS-C4948-10GE
S/N: FOX1203G1XG Hw: 3.4) is online
*Mar 29 20:19:16 PDT: %SYS-5-CONFIG_I: Configured from memory by console
*Mar 29 20:19:16 PDT: %SYS-5-RESTART: System restarted --
Cisco IOS Software, Catalyst 4500 L3 Switch Software (cat4500-IPBASEK9-M), Version
12.2(50)SG, RELEASE SOFTWARE (fc4)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2008 by Cisco Systems, Inc.
Compiled Wed 24-Dec-08 18:23 by prod_rel_team
*Mar 29 20:19:16 PDT: %IP_SNMP-3-SOCKET: can't open UDP socket
*Mar 29 20:19:16 PDT: Unable to open socket on port 161
*Mar 29 20:19:18 PDT: %C4K_IOSINTF-5-TRANSCEIVERINSERTED: Slot=1 Port=49:
Transceiver has been insertedInterface Vlan1 assigned DHCP address 192.168.100.232,
mask 255.255.255.0
*Mar 29 20:21:16 PDT: %SSH-5-ENABLED: SSH 1.99 has been enabled
*Mar 29 20:21:16 PDT: %HA_EM-6-LOG: key_gen: Crypto key generation has completed.
*Mar 29 20:21:16 PDT: %SYS-5-CONFIG_I: Configured from console by vty0
C4948-36>en
C4948-36#dir bo
C4948-36#dir bootflash:
Directory of bootflash:/
    1  -rwx    15155232  Aug 31 2009 11:07:48 -07:00  cat4500-ipbasek9-mz.122-46.SG
    2  -rw-       12624  Mar 29 2010 20:14:59 -07:00  autoconf.tcl
    3  -rwx    15728024  Mar 29 2010 20:15:22 -07:00  cat4500-ipbasek9-mz.122-50.SG
    4  -rw-         761  Mar 29 2010 20:17:54 -07:00  tor.cfg
61341696 bytes total (30444540 bytes free)
C4948-36#sho run
Building configuration...
Current configuration : 2963 bytes
!
version 12.2
no service pad
service timestamps debug datetime localtime show-timezone
service timestamps log datetime localtime show-timezone
service password-encryption
```

```
service compress-config
!
hostname C4948-36
!
boot-start-marker
boot system bootflash:cat4500-ipbasek9-mz.122-50.SG
boot-end-marker
!
!
no aaa new-model
clock timezone PST -8
clock summer-time PDT recurring
ip subnet-zero
ip domain-name cisco.com
!
ip vrf mgmtVrf
!
vtp mode transparent
!
!
!
system mtu 1552
power redundancy-mode redundant
!
!
!
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
no spanning-tree vlan 1-4094
!
vlan internal allocation policy ascending
!
vlan 100
!
!
!
interface FastEthernet1
 ip vrf forwarding mgmtVrf
 no ip address
 shutdown
 speed auto
 duplex auto
!
interface GigabitEthernet1/1
!
interface GigabitEthernet1/2
!
```

```
interface GigabitEthernet1/3
!
interface GigabitEthernet1/4
!
interface GigabitEthernet1/5
!
interface GigabitEthernet1/6
!
interface GigabitEthernet1/7
!
interface GigabitEthernet1/8
!
interface GigabitEthernet1/9
!
interface GigabitEthernet1/10
!
interface GigabitEthernet1/11
!
interface GigabitEthernet1/12
!
interface GigabitEthernet1/13
!
interface GigabitEthernet1/14
!
interface GigabitEthernet1/15
!
interface GigabitEthernet1/16
!
interface GigabitEthernet1/17
!
interface GigabitEthernet1/18
!
interface GigabitEthernet1/19
!
interface GigabitEthernet1/20
!
interface GigabitEthernet1/21
!
interface GigabitEthernet1/22
!
interface GigabitEthernet1/23
!
interface GigabitEthernet1/24
!
interface GigabitEthernet1/25
!
interface GigabitEthernet1/26
!
interface GigabitEthernet1/27
!
```

```
interface GigabitEthernet1/28
!
interface GigabitEthernet1/29
!
interface GigabitEthernet1/30
!
interface GigabitEthernet1/31
!
interface GigabitEthernet1/32
!
interface GigabitEthernet1/33
!
interface GigabitEthernet1/34
!
interface GigabitEthernet1/35
!
interface GigabitEthernet1/36
!
interface GigabitEthernet1/37
!
interface GigabitEthernet1/38
!
interface GigabitEthernet1/39
!
interface GigabitEthernet1/40
!
interface GigabitEthernet1/41
!
interface GigabitEthernet1/42
!
interface GigabitEthernet1/43
!
interface GigabitEthernet1/44
!
interface GigabitEthernet1/45
!
interface GigabitEthernet1/46
!
interface GigabitEthernet1/47
!
interface GigabitEthernet1/48
!
interface TenGigabitEthernet1/49
!
interface TenGigabitEthernet1/50
!
interface Vlan1
 ip address dhcp
!
ip http server
```

```
no ip http secure-server
!
!
!
snmp-server user labRW labRW v1 access 2
snmp-server user labRW labRW v2c access 2
snmp-server community labRO RO 2
snmp-server community labRW RW 2
snmp-server location SJC-North
!
control-plane
!
!
line con 0
 stopbits 1
line vty 0 4
 login
 transport preferred none
 transport input ssh
line vty 5 15
 login
 transport preferred none
 transport input ssh
!
event manager environment _autoconf_url http://192.168.100.186/cgi-
bin/autoinstall/autoconf.cgi
event manager directory user policy "bootflash:/"
end
C4948-36#
C4948-36#
```

Appendix F: Console Logs When the System Is Running the Correct Image

```
switch#dir bootflash:
Directory of bootflash:/
    1  -rwx    15155232  Aug 31 2009 11:07:48 -07:00  cat4500-ipbasek9-mz.122-46.SG
    3  -rwx    15728024  Mar 29 2010 20:15:22 -07:00  cat4500-ipbasek9-mz.122-50.SG
61341696 bytes total (30444540 bytes free)
switch#sho bo
switch#sho bootv
switch#sho bootvar
BOOT variable = bootflash:cat4500-ipbasek9-mz.122-50.SG,12;
CONFIG_FILE variable =
BOOTLDR variable does not exist
Configuration register is 0x2102
switch#reload
System configuration has been modified. Save? [yes/no]: no
Proceed with reload? [confirm]
*Mar 29 20:25:05 PDT: %SYS-5-RELOAD: Reload requested by console. Reload reason:
Reload command
************************************************************
```

```
*                                                          *
* Welcome to Rom Monitor for WS-C4948-10GE System.        *
* Copyright (c) 2003-2006 by Cisco Systems, Inc.          *
* All rights reserved.                                     *
*                                                          *
************************************************************
 Rom Monitor Program Version 12.2(31r)SGA1
 Supervisor: WS-C4948-10GE  Chassis: WS-C4948
 Hardware Revisions - Board: 8.5 CPLD Gill: 20
 MAC Address  : 00-1f-9e-af-cc-3f
 Ip Address   : Not set.
 Netmask      : Not set.
 Gateway      : Not set.
 TftpServer   : Not set.
 ***** The system will autoboot in 5 seconds *****
 Type control-C to prevent autobooting.
 . . . . .
 Established physical link 1Gb Full Duplex
 Network layer connectivity may take a few seconds
 ******** The system will autoboot now ********
 config-register = 0x2102
 Autobooting using BOOT variable specified file.....
 Current BOOT file is --- bootflash:cat4500-ipbasek9-mz.122-50.SG
Rommon reg: 0x00004180
#############
k2diags version 5.2_d
prod: WS-C4948-10GE  part: 73-9344-10  serial: FOX1203G1XG
Power-on-self-test for Module 1:  WS-C4948-10GE
 Port/Test Status: (. = Pass, F = Fail, U = Untested)
Cpu Subsystem Tests ...
seeprom: . temperature_sensor: .
Port Traffic: L3 Serdes Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .
62: . 63: .
Port Traffic: L2 Serdes Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .
62: . 63: .
Port Traffic: L2 Asic Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .
62: . 63: .
Switch Subsystem Memory ...
```

```
 1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: . 12: .
13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: . 24: .
25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: . 36: .
37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: . 48: .
49: . 50: . 51: .
Module 1 Passed
Exiting to ios...
Rommon reg: 0x00000180
############################################
              Restricted Rights Legend
Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.
           cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706
Cisco IOS Software, Catalyst 4500 L3 Switch Software (cat4500-IPBASEK9-M), Version
12.2(50)SG, RELEASE SOFTWARE (fc4)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2008 by Cisco Systems, Inc.
Compiled Wed 24-Dec-08 18:23 by prod_rel_team
Image text-base: 0x10000000, data-base: 0x11C2928C
This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.
A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html
If you require further assistance please contact us by sending email to
export@cisco.com.
cisco WS-C4948-10GE (MPC8540) processor (revision 5) with 262144K bytes of memory.
Processor board ID FOX1203G1XG
MPC8540 CPU at 667Mhz, Fixed Module
Last reset from Reload
1 Virtual Ethernet interface
48 Gigabit Ethernet interfaces
2 Ten Gigabit Ethernet interfaces
511K bytes of non-volatile configuration memory.
Press RETURN to get started!
*Mar  1 00:00:01.347: %C4K_IOSSYS-3-BLANKSTARTUPCONFIG: Blank or invalid startup-
config, booting up with defaults
*Mar 30 03:26:12.547: %SPANTREE-5-EXTENDED_SYSID: Extended SysId enabled for type
vlan
```

```
*Mar 30 03:26:14.435: %C4K_IOSMODPORTMAN-6-MODULEONLINE: Module 1 (WS-C4948-10GE
S/N: FOX1203G1XG Hw: 3.4) is online

*Mar 30 03:26:33.615: %SYS-5-RESTART: System restarted --

Cisco IOS Software, Catalyst 4500 L3 Switch Software (cat4500-IPBASEK9-M), Version
12.2(50)SG, RELEASE SOFTWARE (fc4)

Technical Support: http://www.cisco.com/techsupport

Copyright (c) 1986-2008 by Cisco Systems, Inc.

Compiled Wed 24-Dec-08 18:23 by prod_rel_team

*Mar 30 03:26:35.383: %SPANTREE-7-RECV_1Q_NON_TRUNK: Received 802.1Q BPDU on non
trunk TenGigabitEthernet1/49 VLAN1.

*Mar 30 03:26:35.383: %SPANTREE-7-BLOCK_PORT_TYPE: Blocking TenGigabitEthernet1/49
on VLAN0001. Inconsistent port type.

*Mar 30 03:27:35.443: AUTOINSTALL: Vlan1 is assigned 192.168.100.233

*Mar 30 03:27:45.443: AUTOINSTALL: Obtain tftp server address (opt 150)
192.168.100.186

Loading c4948_initial.cfg from 192.168.100.186 (via Vlan1): !

[OK - 1004 bytes]

*Mar 30 03:27:58.495: %SYS-5-CONFIG_I: Configured from
tftp://192.168.100.186/c4948_initial.cfg by console

*Mar 30 03:28:58.531: %SYS-5-CONFIG_I: Configured from console by  on vty0
(EEM:copy_script)

*Mar 30 03:28:59.783: %HA_EM-6-LOG: copy_script: done with tc autoconf copy

*Mar 30 03:28:59.935: %HA_EM-6-LOG: copy_script: done with registering autoconf tcl
script

*Mar 30 03:28:59.947: %SYS-5-CONFIG_I: Configured from console by  on vty0
(EEM:copy_script)

*Mar 30 03:29:00.547: %HA_EM-6-LOG: copy_script: remove myself and save the
configuration

*Mar 30 03:29:15.079: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_open
called.

*Mar 30 03:29:15.211: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch>

*Mar 30 03:29:15.211: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch>enable

*Mar 30 03:29:15.323: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#

*Mar 30 03:29:15.323: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#show ip int brief | include up                    up

*Mar 30 03:29:15.435: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Vlan1
192.168.100.233 YES DHCP   up                    up

*Mar 30 03:29:15.435: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
TenGigabitEthernet1/49 unassigned      YES unset  up                      up

*Mar 30 03:29:15.435: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#

*Mar 30 03:29:15.435: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#show version | include Cisco IOS Software

*Mar 30 03:29:15.547: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Cisco IOS
Software, Catalyst 4500 L3 Switch Software (cat4500-IPBASEK9-M), Version
12.2(50)SG, RELEASE SOFTWARE (fc4)

*Mar 30 03:29:15.547: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#

*Mar 30 03:29:15.547: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#dir bootflash:

*Mar 30 03:29:15.659: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Directory
of bootflash:/

*Mar 30 03:29:15.659: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :

*Mar 30 03:29:15.659: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :     1  -
rwx    15155232  Aug 31 2009 18:07:48 +00:00  cat4500-ipbasek9-mz.122-46.SG

*Mar 30 03:29:15.659: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :     3  -
rwx    15728024  Mar 30 2010 03:15:22 +00:00  cat4500-ipbasek9-mz.122-50.SG
```

```
*Mar 30 03:29:15.659: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :     5  -
rw-       12624  Mar 30 2010 03:28:59 +00:00  autoconf.tcl

*Mar 30 03:29:15.659: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :

*Mar 30 03:29:15.659: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : 61341696
bytes total (30431788 bytes free)

*Mar 30 03:29:15.659: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#

*Mar 30 03:29:15.659: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_close
called.

*Mar 30 03:29:15.799: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_open
called.

*Mar 30 03:29:15.911: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch>

*Mar 30 03:29:15.911: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch>enable

*Mar 30 03:29:16.023: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#

*Mar 30 03:29:16.023: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#configure terminal

*Mar 30 03:29:16.151: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.

*Mar 30 03:29:16.151: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#

*Mar 30 03:29:16.151: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#file prompt quiet

*Mar 30 03:29:16.263: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#

*Mar 30 03:29:16.263: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#end

*Mar 30 03:29:16.263: %SYS-5-CONFIG_I: Configured from console by  on vty0
(EEM:autoconf.tcl)

*Mar 30 03:29:16.375: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#

*Mar 30 03:29:16.375: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#copy tftp://192.168.100.186//images/tor.cfg bootflash:

*Mar 30 03:29:16.587: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Accessing
tftp://192.168.100.186//images/tor.cfg...

*Mar 30 03:29:16.587: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Loading
/images/tor.cfg from 192.168.100.186 (via Vlan1): !

*Mar 30 03:29:16.587: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : [OK - 761
bytes]

*Mar 30 03:29:16.587: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :

*Mar 30 03:29:16.587: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : 761 bytes
copied in 0.024 secs (31708 bytes/sec)

*Mar 30 03:29:16.587: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#

*Mar 30 03:29:16.587: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#configure terminal

*Mar 30 03:29:16.699: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.

*Mar 30 03:29:16.699: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#

*Mar 30 03:29:16.699: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#no file prompt quiet

*Mar 30 03:29:16.811: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#

*Mar 30 03:29:16.811: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#end

*Mar 30 03:29:16.811: %SYS-5-CONFIG_I: Configured from console by  on vty0
(EEM:autoconf.tcl)

*Mar 30 03:29:16.923: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#
```

```
*Mar 30 03:29:16.923: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_close
called.
*Mar 30 03:29:16.923: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_open
called.
*Mar 30 03:29:17.035: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch>
*Mar 30 03:29:17.035: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch>enable
*Mar 30 03:29:17.147: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#
*Mar 30 03:29:17.147: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#configure terminal
*Mar 30 03:29:17.259: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.
*Mar 30 03:29:17.259: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#
*Mar 30 03:29:17.259: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#file prompt quiet
*Mar 30 03:29:17.371: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#
*Mar 30 03:29:17.371: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#end
*Mar 30 03:29:17.371: %SYS-5-CONFIG_I: Configured from console by  on vty1
(EEM:autoconf.tcl)
*Mar 30 03:29:17.483: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#
*Mar 30 03:29:17.483: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#copy tftp://192.168.100.186//images/tor.cfg running
*Mar 29 19:29:26 PST: %SYS-6-CLOCKUPDATE: System clock has been updated from
03:29:26 UTC Tue Mar 30 2010 to 19:29:26 PST Mon Mar 29 2010, configured from
console by  on vty1 (EEM:autoconf.tcl).
*Mar 29 20:29:26 PDT: %SYS-6-CLOCKUPDATE: System clock has been updated from
19:29:26 PST Mon Mar 29 2010 to 20:29:26 PDT Mon Mar 29 2010, configured from
console by  on vty1 (EEM:autoconf.tcl).
*Mar 29 20:29:26 PDT: %SYS-5-CONFIG_I: Configured from
tftp://192.168.100.186//images/tor.cfg by  on vty1 (EEM:autoconf.tcl)
*Mar 29 20:29:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Accessing
tftp://192.168.100.186//images/tor.cfg...
*Mar 29 20:29:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Loading
/images/tor.cfg from 192.168.100.186 (via Vlan1): !
*Mar 29 20:29:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : [OK - 761
bytes]
*Mar 29 20:29:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Global
Ethernet MTU is set to 1552 bytes.
*Mar 29 20:29:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Note:
this is the Ethernet payload size, not the total
*Mar 29 20:29:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Ethernet
frame size, which includes the Ethernet
*Mar 29 20:29:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
header/trailer and possibly other tags, such as ISL or
*Mar 29 20:29:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : 802.1q
tags.
*Mar 29 20:29:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
*Mar 29 20:29:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : 761 bytes
copied in 9.104 secs (84 bytes/sec)
*Mar 29 20:29:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Mar 29 20:29:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
#configure terminal
*Mar 29 20:29:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.
```

```
*Mar 29 20:29:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Mar 29 20:29:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#no file prompt quiet
*Mar 29 20:29:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Mar 29 20:29:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#end
*Mar 29 20:29:26 PDT: %SYS-5-CONFIG_I: Configured from console by  on vty1
(EEM:autoconf.tcl)
*Mar 29 20:29:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Mar 29 20:29:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_close
called.
*Mar 29 20:29:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_open
called.
*Mar 29 20:29:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36>
*Mar 29 20:29:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36>enable
*Mar 29 20:29:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Mar 29 20:29:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36#configure terminal
*Mar 29 20:29:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.
*Mar 29 20:29:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Mar 29 20:29:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#no event manager policy autoconf.tcl
*Mar 29 20:29:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Mar 29 20:29:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#end
*Mar 29 20:29:27 PDT: %SYS-5-CONFIG_I: Configured from console by  on vty0
(EEM:autoconf.tcl)
*Mar 29 20:29:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Mar 29 20:29:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36#config t
*Mar 29 20:29:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.
*Mar 29 20:29:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Mar 29 20:29:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#crypto key zeroize rsa
*Mar 29 20:29:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : % No
Signature RSA Keys found in configuration.
*Mar 29 20:29:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
*Mar 29 20:29:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Mar 29 20:29:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : ip
domain-name cisco.com
*Mar 29 20:29:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Mar 29 20:29:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#crypto key generate rsa
*Mar 29 20:29:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : The name
for the keys will be: C4948-36.cisco.com
*Mar 29 20:29:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Choose
the size of the key modulus in the range of 360 to 2048 for your
```

```
*Mar 29 20:29:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :   General
Purpose Keys. Choosing a key modulus greater than 512 may take
*Mar 29 20:29:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :   a few
minutes.
*Mar 29 20:29:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
*Mar 29 20:29:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : How many
bits in the modulus [512]:
*Mar 29 20:29:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : 1024
*Mar 29 20:29:28 PDT: %SSH-5-ENABLED: SSH 1.99 has been enabled
*Mar 29 20:29:29 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : %
Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
*Mar 29 20:29:29 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
*Mar 29 20:29:29 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Mar 29 20:29:29 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#end
*Mar 29 20:29:29 PDT: %SYS-5-CONFIG_I: Configured from console by  on vty0
(EEM:autoconf.tcl)
*Mar 29 20:29:29 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Mar 29 20:29:29 PDT: %HA_EM-6-LOG: autoconf.tcl: Crypto key generation has
completed.
*Mar 29 20:29:49 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36#copy run start
*Mar 29 20:29:49 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Destination filename [startup-config]?
*Mar 29 20:29:49 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
*Mar 29 20:29:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Building
configuration...
*Mar 29 20:29:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Compressed configuration from 2828 bytes to 1283 bytes[OK]
*Mar 29 20:29:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Mar 29 20:29:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Mar 29 20:29:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36#configure terminal
*Mar 29 20:29:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.
*Mar 29 20:29:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Mar 29 20:29:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#ev man app reload_helper
*Mar 29 20:29:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Mar 29 20:29:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#ev app sub 798 type 1
*Mar 29 20:29:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Mar 29 20:29:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#action 1 reload
*Mar 29 20:29:51 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Mar 29 20:29:51 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#end
*Mar 29 20:29:51 PDT: %SYS-5-CONFIG_I: Configured from console by  on vty0
(EEM:autoconf.tcl)
*Mar 29 20:29:51 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Mar 29 20:29:51 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_close
called.
```

```
*Mar 29 20:29:51 PDT: %HA_EM-6-LOG: autoconf.tcl: field: server  value:
192.168.100.186
*Mar 29 20:29:51 PDT: %HA_EM-6-LOG: autoconf.tcl: field: action  value: none
*Mar 29 20:29:51 PDT: %HA_EM-6-LOG: autoconf.tcl: field: image  value: null
*Mar 29 20:29:51 PDT: %HA_EM-6-LOG: autoconf.tcl: field: config  value:
/images/tor.cfg
*Mar 29 20:29:51 PDT: %HA_EM-6-LOG: autoconf.tcl: No need to copy the image
*Mar 29 20:29:51 PDT: %HA_EM-6-LOG: autoconf.tcl: start copying configuration file
*Mar 29 20:29:51 PDT: %HA_EM-6-LOG: autoconf.tcl: done copying configuration file
C4948-36>en
C4948-36#dir bootflash:
Directory of bootflash:/
    1  -rwx    15155232  Aug 31 2009 11:07:48 -07:00  cat4500-ipbasek9-mz.122-46.SG
    3  -rwx    15728024  Mar 29 2010 20:15:22 -07:00  cat4500-ipbasek9-mz.122-50.SG
    5  -rw-       12624  Mar 29 2010 20:28:59 -07:00  autoconf.tcl
    6  -rw-         761  Mar 29 2010 20:29:16 -07:00  tor.cfg
61341696 bytes total (30430896 bytes free)
C4948-36#sho run
Building configuration...
Current configuration : 2924 bytes
!
version 12.2
no service pad
service timestamps debug datetime localtime show-timezone
service timestamps log datetime localtime show-timezone
service password-encryption
service compress-config
!
hostname C4948-36
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
clock timezone PST -8
clock summer-time PDT recurring
ip subnet-zero
ip domain-name cisco.com
!
ip vrf mgmtVrf
!
vtp mode transparent
!
!
!
system mtu 1552
power redundancy-mode redundant
!
!
```

```
!
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
no spanning-tree vlan 1-4094
!
vlan internal allocation policy ascending
!
vlan 100
!
!
!
interface FastEthernet1
 ip vrf forwarding mgmtVrf
 no ip address
 shutdown
 speed auto
 duplex auto
!
interface GigabitEthernet1/1
!
interface GigabitEthernet1/2
!
interface GigabitEthernet1/3
!
interface GigabitEthernet1/4
!
interface GigabitEthernet1/5
!
interface GigabitEthernet1/6
!
interface GigabitEthernet1/7
!
interface GigabitEthernet1/8
!
interface GigabitEthernet1/9
!
interface GigabitEthernet1/10
!
interface GigabitEthernet1/11
!
interface GigabitEthernet1/12
!
interface GigabitEthernet1/13
!
interface GigabitEthernet1/14
!
interface GigabitEthernet1/15
```

```
!
interface GigabitEthernet1/16
!
interface GigabitEthernet1/17
!
interface GigabitEthernet1/18
!
interface GigabitEthernet1/19
!
interface GigabitEthernet1/20
!
interface GigabitEthernet1/21
!
interface GigabitEthernet1/22
!
interface GigabitEthernet1/23
!
interface GigabitEthernet1/24
!
interface GigabitEthernet1/25
!
interface GigabitEthernet1/26
!
interface GigabitEthernet1/27
!
interface GigabitEthernet1/28
!
interface GigabitEthernet1/29
!
interface GigabitEthernet1/30
!
interface GigabitEthernet1/31
!
interface GigabitEthernet1/32
!
interface GigabitEthernet1/33
!
interface GigabitEthernet1/34
!
interface GigabitEthernet1/35
!
interface GigabitEthernet1/36
!
interface GigabitEthernet1/37
!
interface GigabitEthernet1/38
!
interface GigabitEthernet1/39
!
interface GigabitEthernet1/40
```

```
!
interface GigabitEthernet1/41
!
interface GigabitEthernet1/42
!
interface GigabitEthernet1/43
!
interface GigabitEthernet1/44
!
interface GigabitEthernet1/45
!
interface GigabitEthernet1/46
!
interface GigabitEthernet1/47
!
interface GigabitEthernet1/48
!
interface TenGigabitEthernet1/49
!
interface TenGigabitEthernet1/50
!
interface Vlan1
 ip address dhcp
!
ip http server
no ip http secure-server
!
!
!
snmp-server community labRO RO 2
snmp-server community labRW RW 2
snmp-server location SJC-North
!
control-plane
!
!
line con 0
 stopbits 1
line vty 0 4
 login
 transport preferred none
 transport input ssh
line vty 5 15
 login
 transport preferred none
 transport input ssh
!
event manager environment _autoconf_url http://192.168.100.186/cgi-
bin/autoinstall/autoconf.cgi
event manager directory user policy "bootflash:/"
```

```
event manager applet reload_helper
 event application sub-system 798 type 1
 action 1 reload
!
end
C4948-36#
```

Appendix G: Console Logs When the Required Image Is on the Local File System, but the File System Is Not Running It

```
Switch#show bootvar
BOOT variable =
CONFIG_FILE variable =
BOOTLDR variable does not exist
Configuration register is 0x2101
Switch#dir bootflash:
Directory of bootflash:/
    1  -rwx    15155232  Aug 31 2009 18:07:48 +00:00  cat4500-ipbasek9-mz.122-46.SG
   10  -rwx    15728024  Mar 30 2010 17:25:22 +00:00  cat4500-ipbasek9-mz.122-50.SG
61341696 bytes total (14675456 bytes free)
Switch#reload
Proceed with reload? [confirm]
*Apr 18 21:20:43.725: %SYS-5-RELOAD: Reload requested by console. Reload reason:
Reload command
************************************************************
*                                                          *
* Welcome to Rom Monitor for WS-C4948-10GE System.         *
* Copyright (c) 2003-2006 by Cisco Systems, Inc.           *
* All rights reserved.                                     *
*                                                          *
************************************************************
Rom Monitor Program Version 12.2(31r)SGA1
 Supervisor: WS-C4948-10GE  Chassis: WS-C4948
 Hardware Revisions - Board: 8.5 CPLD Gill: 20
 MAC Address  : 00-1f-9e-af-cc-3f
 Ip Address   : Not set.
 Netmask      : Not set.
 Gateway      : Not set.
 TftpServer   : Not set.
***** The system will autoboot in 5 seconds *****
Type control-C to prevent autobooting.
 . . . . .
 Established physical link 1Gb Full Duplex
 Network layer connectivity may take a few seconds
 ******** The system will autoboot now ********
 config-register = 0x2101
 Autobooting using the first file from bootflash.....
Rommon reg: 0x00004180
#############
k2diags version 5.2_c
prod: WS-C4948-10GE  part: 73-9344-10  serial: FOX1203G1XG
```

```
Power-on-self-test for Module 1:  WS-C4948-10GE
 Port/Test Status: (. = Pass, F = Fail, U = Untested)
Cpu Subsystem Tests ...
seeprom: . temperature_sensor: .
Port Traffic: L3 Serdes Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .
62: . 63: .
Port Traffic: L2 Serdes Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .
62: . 63: .
Port Traffic: L2 Asic Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .
62: . 63: .
Switch Subsystem Memory ...
 1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: . 12: .
13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: . 24: .
25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: . 36: .
37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: . 48: .
49: . 50: . 51: .
Module 1 Passed
Exiting to ios...
Rommon reg: 0x00000180
########################################
              Restricted Rights Legend
Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.
           cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706
Cisco IOS Software, Catalyst 4500 L3 Switch Software (cat4500-IPBASEK9-M), Version
12.2(46)SG, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2008 by Cisco Systems, Inc.
Compiled Fri 27-Jun-08 16:56 by prod_rel_team
Image text-base: 0x10000000, data-base: 0x11ABEC24
This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
```

```
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.
A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html
If you require further assistance please contact us by sending email to
export@cisco.com.
cisco WS-C4948-10GE (MPC8540) processor (revision 5) with 262144K bytes of memory.
Processor board ID FOX1203G1XG
MPC8540 CPU at 667Mhz, Fixed Module
Last reset from Reload
1 Virtual Ethernet interface
48 Gigabit Ethernet interfaces
2 Ten Gigabit Ethernet interfaces
511K bytes of non-volatile configuration memory.
Press RETURN to get started!
*Mar  1 00:00:01.327: %C4K_IOSSYS-3-BLANKSTARTUPCONFIG: Blank or invalid startup-
config, booting up with defaults
*Apr 18 21:22:12.531: %SPANTREE-5-EXTENDED_SYSID: Extended SysId enabled for type
vlan
*Apr 18 21:22:14.443: %C4K_IOSMODPORTMAN-6-MODULEONLINE: Module 1 (WS-C4948-10GE
S/N: FOX1203G1XG Hw: 3.4) is online
*Apr 18 21:22:33.611: %SYS-5-RESTART: System restarted --
Cisco IOS Software, Catalyst 4500 L3 Switch Software (cat4500-IPBASEK9-M), Version
12.2(46)SG, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2008 by Cisco Systems, Inc.
Compiled Fri 27-Jun-08 16:56 by prod_rel_team
*Apr 18 21:22:35.791: %SPANTREE-7-RECV_1Q_NON_TRUNK: Received 802.1Q BPDU on non
trunk TenGigabitEthernet1/49 VLAN1.
*Apr 18 21:22:35.791: %SPANTREE-7-BLOCK_PORT_TYPE: Blocking TenGigabitEthernet1/49
on VLAN0001. Inconsistent port type.
*Apr 18 21:23:35.443: AUTOINSTALL: Vlan1 is assigned 192.168.100.238
*Apr 18 21:23:45.443: AUTOINSTALL: Obtain tftp server address (opt 150)
192.168.100.186
Loading c4948_initial.cfg from 192.168.100.186 (via Vlan1): !
[OK - 1004 bytes]
*Apr 18 21:23:58.491: %SYS-5-CONFIG_I: Configured from
tftp://192.168.100.186/c4948_initial.cfg by console
*Apr 18 21:24:58.531: %SYS-5-CONFIG_I: Configured from console by  on vty0
(EEM:copy_script)
*Apr 18 21:24:59.787: %HA_EM-6-LOG: copy_script: done with tc autoconf copy
*Apr 18 21:24:59.935: %HA_EM-6-LOG: copy_script: done with registering autoconf tcl
script
*Apr 18 21:24:59.947: %SYS-5-CONFIG_I: Configured from console by  on vty0
(EEM:copy_script)
*Apr 18 21:25:00.515: %HA_EM-6-LOG: copy_script: remove myself and save the
configuration
*Apr 18 21:25:15.075: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_open
called.
*Apr 18 21:25:15.207: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch>
```

```
*Apr 18 21:25:15.207: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch>enable
*Apr 18 21:25:15.319: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#
*Apr 18 21:25:15.319: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#show ip int brief | include up                    up
*Apr 18 21:25:15.431: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Vlan1
192.168.100.238 YES DHCP   up                    up
*Apr 18 21:25:15.431: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
TenGigabitEthernet1/49 unassigned      YES unset  up                    up
*Apr 18 21:25:15.431: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#
*Apr 18 21:25:15.431: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#show version | include Cisco IOS Software
*Apr 18 21:25:15.543: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Cisco IOS
Software, Catalyst 4500 L3 Switch Software (cat4500-IPBASEK9-M), Version
12.2(46)SG, RELEASE SOFTWARE (fc1)
*Apr 18 21:25:15.543: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#
*Apr 18 21:25:15.543: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#dir bootflash:
*Apr 18 21:25:15.755: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Directory
of bootflash:/
*Apr 18 21:25:15.755: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
*Apr 18 21:25:15.755: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :     1  -
rwx    15155232  Aug 31 2009 18:07:48 +00:00  cat4500-ipbasek9-mz.122-46.SG
*Apr 18 21:25:15.755: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :    10  -
rwx    15728024  Mar 30 2010 17:25:22 +00:00  cat4500-ipbasek9-mz.122-50.SG
*Apr 18 21:25:15.755: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :    12  -
rw-       12624  Apr 18 2010 21:24:59 +00:00  autoconf.tcl
*Apr 18 21:25:15.755: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
*Apr 18 21:25:15.755: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : 61341696
bytes total (14662704 bytes free)
*Apr 18 21:25:15.755: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#
*Apr 18 21:25:15.755: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_close
called.
*Apr 18 21:25:15.895: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_open
called.
*Apr 18 21:25:16.007: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch>
*Apr 18 21:25:16.007: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch>enable
*Apr 18 21:25:16.119: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#
*Apr 18 21:25:16.119: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#configure terminal
*Apr 18 21:25:16.347: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.
*Apr 18 21:25:16.347: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#
*Apr 18 21:25:16.347: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#file prompt quiet
*Apr 18 21:25:16.459: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#
*Apr 18 21:25:16.459: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#end
*Apr 18 21:25:16.459: %SYS-5-CONFIG_I: Configured from console by  on vty0
(EEM:autoconf.tcl)
*Apr 18 21:25:16.571: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#
*Apr 18 21:25:16.571: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#copy tftp://192.168.100.186//images/tor.cfg bootflash:
```

```
*Apr 18 21:25:16.783: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Accessing
tftp://192.168.100.186//images/tor.cfg...

*Apr 18 21:25:16.783: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Loading
/images/tor.cfg from 192.168.100.186 (via Vlan1): !

*Apr 18 21:25:16.783: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : [OK - 761
bytes]

*Apr 18 21:25:16.783: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :

*Apr 18 21:25:16.783: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : 761 bytes
copied in 0.020 secs (38050 bytes/sec)

*Apr 18 21:25:16.783: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#

*Apr 18 21:25:16.783: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#configure terminal

*Apr 18 21:25:16.895: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.

*Apr 18 21:25:16.895: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#

*Apr 18 21:25:16.895: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#no file prompt quiet

*Apr 18 21:25:17.007: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#

*Apr 18 21:25:17.007: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#end

*Apr 18 21:25:17.007: %SYS-5-CONFIG_I: Configured from console by  on vty0
(EEM:autoconf.tcl)

*Apr 18 21:25:17.119: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#

*Apr 18 21:25:17.119: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_close
called.

*Apr 18 21:25:17.119: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_open
called.

*Apr 18 21:25:17.231: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch>

*Apr 18 21:25:17.231: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch>enable

*Apr 18 21:25:17.343: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#

*Apr 18 21:25:17.343: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#configure terminal

*Apr 18 21:25:17.455: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.

*Apr 18 21:25:17.455: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#

*Apr 18 21:25:17.455: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#file prompt quiet

*Apr 18 21:25:17.567: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Switch(config)#

*Apr 18 21:25:17.567: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch(config)#end

*Apr 18 21:25:17.567: %SYS-5-CONFIG_I: Configured from console by  on vty1
(EEM:autoconf.tcl)

*Apr 18 21:25:17.679: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Switch#

*Apr 18 21:25:17.679: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
Switch#copy tftp://192.168.100.186//images/tor.cfg running

*Apr 18 13:25:26 PST: %SYS-6-CLOCKUPDATE: System clock has been updated from
21:25:26 UTC Sun Apr 18 2010 to 13:25:26 PST Sun Apr 18 2010, configured from
console by  on vty1 (EEM:autoconf.tcl).

*Apr 18 14:25:26 PDT: %SYS-6-CLOCKUPDATE: System clock has been updated from
13:25:26 PST Sun Apr 18 2010 to 14:25:26 PDT Sun Apr 18 2010, configured from
console by  on vty1 (EEM:autoconf.tcl).

*Apr 18 14:25:26 PDT: %SYS-5-CONFIG_I: Configured from
tftp://192.168.100.186//images/tor.cfg by  on vty1 (EEM:autoconf.tcl)
```

```
*Apr 18 14:25:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Accessing
tftp://192.168.100.186//images/tor.cfg...
*Apr 18 14:25:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Loading
/images/tor.cfg from 192.168.100.186 (via Vlan1): !
*Apr 18 14:25:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : [OK - 761
bytes]
*Apr 18 14:25:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Global
Ethernet MTU is set to 1552 bytes.
*Apr 18 14:25:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Note:
this is the Ethernet payload size, not the total
*Apr 18 14:25:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Ethernet
frame size, which includes the Ethernet
*Apr 18 14:25:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
header/trailer and possibly other tags, such as ISL or
*Apr 18 14:25:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : 802.1q
tags.
*Apr 18 14:25:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
*Apr 18 14:25:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : 761 bytes
copied in 9.104 secs (84 bytes/sec)
*Apr 18 14:25:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Apr 18 14:25:26 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
#configure terminal
*Apr 18 14:25:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.
*Apr 18 14:25:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Apr 18 14:25:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#no file prompt quiet
*Apr 18 14:25:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Apr 18 14:25:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#end
*Apr 18 14:25:27 PDT: %SYS-5-CONFIG_I: Configured from console by  on vty1
(EEM:autoconf.tcl)
*Apr 18 14:25:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Apr 18 14:25:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_close
called.
*Apr 18 14:25:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_open
called.
*Apr 18 14:25:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36>
*Apr 18 14:25:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36>enable
*Apr 18 14:25:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Apr 18 14:25:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36#configure terminal
*Apr 18 14:25:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.
*Apr 18 14:25:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Apr 18 14:25:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#no event manager policy autoconf.tcl
*Apr 18 14:25:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Apr 18 14:25:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#no boot system
*Apr 18 14:25:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
```

```
*Apr 18 14:25:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#boot system bootflash:cat4500-ipbasek9-mz.122-50.SG
*Apr 18 14:25:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Apr 18 14:25:27 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#config-register 0x2102
*Apr 18 14:25:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Apr 18 14:25:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#end
*Apr 18 14:25:28 PDT: %SYS-5-CONFIG_I: Configured from console by  on vty0
(EEM:autoconf.tcl)
*Apr 18 14:25:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Apr 18 14:25:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36#config t
*Apr 18 14:25:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.
*Apr 18 14:25:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Apr 18 14:25:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#ip domain-name cisco.com
*Apr 18 14:25:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Apr 18 14:25:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config)#ev man app key_gen
*Apr 18 14:25:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Apr 18 14:25:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#ev timer countdown time 120 name key maxrun 300
*Apr 18 14:25:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Apr 18 14:25:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#action 1 cli command "enable"
*Apr 18 14:25:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Apr 18 14:25:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#action 2 cli command "config t"
*Apr 18 14:25:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Apr 18 14:25:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#action 3 cli command "crypto key generate rsa" pattern "How many
bits in the modulus"
*Apr 18 14:25:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Apr 18 14:25:28 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#action 4 cli command "1024"
*Apr 18 14:25:29 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Apr 18 14:25:29 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#action 5 syslog msg "Crypto key generation has completed."
*Apr 18 14:25:29 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Apr 18 14:25:29 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#action 6 cli command "no ev man app key_gen"
*Apr 18 14:25:29 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Apr 18 14:25:29 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#end
```

```
*Apr 18 14:25:29 PDT: %SYS-5-CONFIG_I: Configured from console by  on vty0
(EEM:autoconf.tcl)
*Apr 18 14:25:29 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Apr 18 14:25:49 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36#copy run start
*Apr 18 14:25:49 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Destination filename [startup-config]?
*Apr 18 14:25:49 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  :
*Apr 18 14:25:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Building
configuration...
*Apr 18 14:25:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT :
Compressed configuration from 3333 bytes to 1582 bytes[OK]
*Apr 18 14:25:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Apr 18 14:25:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36#configure terminal
*Apr 18 14:25:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Apr 18 14:25:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : Enter
configuration commands, one per line.  End with CNTL/Z.
*Apr 18 14:25:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config)#
*Apr 18 14:25:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36#ev man app reload_helper
*Apr 18 14:25:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Apr 18 14:25:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#ev app sub 798 type 1
*Apr 18 14:25:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Apr 18 14:25:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#action 1 reload
*Apr 18 14:25:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-
36(config-applet)#
*Apr 18 14:25:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : IN  : C4948-
36(config-applet)#end
*Apr 18 14:25:50 PDT: %SYS-5-CONFIG_I: Configured from console by  on vty0
(EEM:autoconf.tcl)
*Apr 18 14:25:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : OUT : C4948-36#
*Apr 18 14:25:50 PDT: %HA_EM-6-LOG: autoconf.tcl : DEBUG(cli_lib) : CTL : cli_close
called.
*Apr 18 14:26:00 PDT: %SYS-5-RELOAD: Reload requested by EEM. Reload reason:
Embedded Event Manager action
**********************************************************
*                                                        *
* Welcome to Rom Monitor for WS-C4948-10GE System.       *
* Copyright (c) 2003-2006 by Cisco Systems, Inc.         *
* All rights reserved.                                   *
*                                                        *
**********************************************************
Rom Monitor Program Version 12.2(31r)SGA1
 Supervisor: WS-C4948-10GE  Chassis: WS-C4948
 Hardware Revisions - Board: 8.5 CPLD Gill: 20
MAC Address  : 00-1f-9e-af-cc-3f
 Ip Address   : Not set.
 Netmask      : Not set.
 Gateway      : Not set.
```

```
 TftpServer   : Not set.
***** The system will autoboot in 5 seconds *****
Type control-C to prevent autobooting.
 . . . . .
 Established physical link 1Gb Full Duplex
 Network layer connectivity may take a few seconds
 ******** The system will autoboot now ********
 config-register = 0x2102
 Autobooting using BOOT variable specified file.....
 Current BOOT file is --- bootflash:cat4500-ipbasek9-mz.122-50.SG
Rommon reg: 0x00004180
#############
k2diags version 5.2_d
prod: WS-C4948-10GE  part: 73-9344-10  serial: FOX1203G1XG
Power-on-self-test for Module 1:  WS-C4948-10GE
 Port/Test Status: (. = Pass, F = Fail, U = Untested)
Cpu Subsystem Tests ...
seeprom: . temperature_sensor: .
Port Traffic: L3 Serdes Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .
62: . 63: .
Port Traffic: L2 Serdes Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .
62: . 63: .
Port Traffic: L2 Asic Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .
62: . 63: .
Switch Subsystem Memory ...
 1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: . 12: .
13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: . 24: .
25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: . 36: .
37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: . 48: .
49: . 50: . 51: .
Module 1 Passed
Exiting to ios...
Rommon reg: 0x00000180
##########################################
             Restricted Rights Legend
Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
```

```
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.
           cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706
Cisco IOS Software, Catalyst 4500 L3 Switch Software (cat4500-IPBASEK9-M), Version
12.2(50)SG, RELEASE SOFTWARE (fc4)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2008 by Cisco Systems, Inc.
Compiled Wed 24-Dec-08 18:23 by prod_rel_team
Image text-base: 0x10000000, data-base: 0x11C2928C
This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.
A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html
If you require further assistance please contact us by sending email to
export@cisco.com.
cisco WS-C4948-10GE (MPC8540) processor (revision 5) with 262144K bytes of memory.
Processor board ID FOX1203G1XG
MPC8540 CPU at 667Mhz, Fixed Module
Last reset from Reload
1 Virtual Ethernet interface
48 Gigabit Ethernet interfaces
2 Ten Gigabit Ethernet interfaces
511K bytes of non-volatile configuration memory.
Press RETURN to get started!
*Apr 18 21:27:12.539: %SPANTREE-5-EXTENDED_SYSID: Extended SysId enabled for type
vlan
*Apr 18 13:27:14 PST: %SYS-6-CLOCKUPDATE: System clock has been updated from
21:27:14 UTC Sun Apr 18 2010 to 13:27:14 PST Sun Apr 18 2010, configured from
console by console.
*Apr 18 14:27:14 PDT: %SYS-6-CLOCKUPDATE: System clock has been updated from
13:27:14 PST Sun Apr 18 2010 to 14:27:14 PDT Sun Apr 18 2010, configured from
console by console.
*Apr 18 14:27:14 PDT: %C4K_IOSMODPORTMAN-6-MODULEONLINE: Module 1 (WS-C4948-10GE
S/N: FOX1203G1XG Hw: 3.4) is online
*Apr 18 14:27:16 PDT: %SYS-5-CONFIG_I: Configured from memory by console
*Apr 18 14:27:16 PDT: %SYS-5-RESTART: System restarted --
Cisco IOS Software, Catalyst 4500 L3 Switch Software (cat4500-IPBASEK9-M), Version
12.2(50)SG, RELEASE SOFTWARE (fc4)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2008 by Cisco Systems, Inc.
Compiled Wed 24-Dec-08 18:23 by prod_rel_team
*Apr 18 14:27:16 PDT: %IP_SNMP-3-SOCKET: can't open UDP socket
*Apr 18 14:27:16 PDT: Unable to open socket on port 161
```

```
*Apr 18 14:27:18 PDT: %C4K_IOSINTF-5-TRANSCEIVERINSERTED: Slot=1 Port=49:
Transceiver has been insertedInterface Vlan1 assigned DHCP address 192.168.100.239,
mask 255.255.255.0
*Apr 18 14:29:17 PDT: %SSH-5-ENABLED: SSH 1.99 has been enabled
*Apr 18 14:29:18 PDT: %HA_EM-6-LOG: key_gen: Crypto key generation has completed.
*Apr 18 14:29:18 PDT: %SYS-5-CONFIG_I: Configured from console by vty0
C4948-36>en
C4948-36#dir bootflash:
Directory of bootflash:/
    1  -rwx    15155232  Aug 31 2009 11:07:48 -07:00  cat4500-ipbasek9-mz.122-46.SG
   10  -rwx    15728024  Mar 30 2010 10:25:22 -07:00  cat4500-ipbasek9-mz.122-50.SG
   12  -rw-       12624  Apr 18 2010 14:24:59 -07:00  autoconf.tcl
   13  -rw-         761  Apr 18 2010 14:25:16 -07:00  tor.cfg
61341696 bytes total (14661812 bytes free)
C4948-36#sho run
Building configuration...
Current configuration : 2963 bytes
!
version 12.2
no service pad
service timestamps debug datetime localtime show-timezone
service timestamps log datetime localtime show-timezone
service password-encryption
service compress-config
!
hostname C4948-36
!
boot-start-marker
boot system bootflash:cat4500-ipbasek9-mz.122-50.SG
boot-end-marker
!
!
no aaa new-model
clock timezone PST -8
clock summer-time PDT recurring
ip subnet-zero
ip domain-name cisco.com
!
ip vrf mgmtVrf
!
vtp mode transparent
!
!
!
system mtu 1552
power redundancy-mode redundant
!
!
!
!
```

```
!
!
spanning-tree mode pvst
spanning-tree extend system-id
no spanning-tree vlan 1-4094
!
vlan internal allocation policy ascending
!
vlan 100
!
!
!
interface FastEthernet1
 ip vrf forwarding mgmtVrf
 no ip address
 shutdown
 speed auto
 duplex auto
!
interface GigabitEthernet1/1
!
interface GigabitEthernet1/2
!
interface GigabitEthernet1/3
!
interface GigabitEthernet1/4
!
interface GigabitEthernet1/5
!
interface GigabitEthernet1/6
!
interface GigabitEthernet1/7
!
interface GigabitEthernet1/8
!
interface GigabitEthernet1/9
!
interface GigabitEthernet1/10
!
interface GigabitEthernet1/11
!
interface GigabitEthernet1/12
!
interface GigabitEthernet1/13
!
interface GigabitEthernet1/14
!
interface GigabitEthernet1/15
!
interface GigabitEthernet1/16
```

```
!
interface GigabitEthernet1/17
!
interface GigabitEthernet1/18
!
interface GigabitEthernet1/19
!
interface GigabitEthernet1/20
!
interface GigabitEthernet1/21
!
interface GigabitEthernet1/22
!
interface GigabitEthernet1/23
!
interface GigabitEthernet1/24
!
interface GigabitEthernet1/25
!
interface GigabitEthernet1/26
!
interface GigabitEthernet1/27
!
interface GigabitEthernet1/28
!
interface GigabitEthernet1/29
!
interface GigabitEthernet1/30
!
interface GigabitEthernet1/31
!
interface GigabitEthernet1/32
!
interface GigabitEthernet1/33
!
interface GigabitEthernet1/34
!
interface GigabitEthernet1/35
!
interface GigabitEthernet1/36
!
interface GigabitEthernet1/37
!
interface GigabitEthernet1/38
!
interface GigabitEthernet1/39
!
interface GigabitEthernet1/40
!
interface GigabitEthernet1/41
```

```
!
interface GigabitEthernet1/42
!
interface GigabitEthernet1/43
!
interface GigabitEthernet1/44
!
interface GigabitEthernet1/45
!
interface GigabitEthernet1/46
!
interface GigabitEthernet1/47
!
interface GigabitEthernet1/48
!
interface TenGigabitEthernet1/49
!
interface TenGigabitEthernet1/50
!
interface Vlan1
 ip address dhcp
!
ip http server
no ip http secure-server
!
!
!
snmp-server user labRW labRW v1 access 2
snmp-server user labRW labRW v2c access 2
snmp-server community labRO RO 2
snmp-server community labRW RW 2
snmp-server location SJC-North
!
control-plane
!
!
line con 0
 stopbits 1
line vty 0 4
 login
 transport preferred none
 transport input ssh
line vty 5 15
 login
 transport preferred none
 transport input ssh
!
event manager environment _autoconf_url http://192.168.100.186/cgi-
bin/autoinstall/autoconf.cgi
event manager directory user policy "bootflash:/"
```

```
end
C4948-36#
```

For More Information

- Cisco Catalyst 4900 Series Switches: http://www.cisco.com/en/US/products/ps6021/index.html.
- Cisco IOS Software AutoInstall and Setup features: http://www.cisco.com/en/US/docs/ios/12_2/configfun/configuration/guide/fcf002.html.
- Cisco IOS Embedded Event Manager: http://www.cisco.com/en/US/products/ps6815/products_ios_protocol_group_home.html.

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

Printed in USA C11-604553-00 06/10