



## CHAPTER 2

# Command-Line Interfaces

---

This chapter describes the CLIs you use to configure the Catalyst 4006 switch with Supervisor Engine III. This chapter includes the following major sections:

- [Accessing the Switch CLI, page 2-2](#)
- [Performing Command-Line Processing, page 2-3](#)
- [Performing History Substitution, page 2-4](#)
- [About Cisco IOS Command Modes, page 2-4](#)
- [Getting a List of Commands and Syntax, page 2-5](#)
- [ROMMON Command-Line Interface, page 2-7](#)
- [Archiving Crashfiles Information, page 2-8](#)
- [Displaying a Crash Dump for Supervisor Engine 6-E and 6L-E, page 2-8](#)



### Note

For complete syntax and usage information for the switch commands used in this chapter, see the *Cisco Catalyst 4500 Series Switch Command Reference* and related publications at this location:  
<http://www.cisco.com/en/US/products/hw/switches/ps4324/index.html>

If a command is not in the *Catalyst 4500 Series Switch Command Reference*, you can locate it in the Cisco IOS library. See related publications at this location:  
<http://www.cisco.com/en/US/products/ps6350/index.html>

The following command changes apply only to Supervisor Engines 6-E and 6L-E:

- The **rename** command is supported in the FAT file system for bootflash and slot0.
- The **fsck** command is supported for the slot0 device. It is not supported in the file systems on supervisor engines other than Supervisor Engine 6-E and 6L-E.

The following additional file management commands are supported on Supervisor Engine 7-E, Supervisor Engine 7L-E, and Supervisor Engine 8-E:

- **verify** <filename>
- **delete** <filename>
- **copy** <source\_filename>, target\_filename>

# Accessing the Switch CLI

The following sections describe how to access the switch CLI:

- [Accessing the CLI Using the EIA/TIA-232 Console Interface, page 2-2](#)
- [Accessing the CLI Through Telnet, page 2-2](#)

## Accessing the CLI Using the EIA/TIA-232 Console Interface



### Note

EIA/TIA-232 was known as recommended standard 232 (RS-232) before its acceptance as a standard by the Electronic Industries Alliance (EIA) and Telecommunications Industry Association (TIA).

Perform the initial switch configuration over a connection to the EIA/TIA-232 console interface. Refer to the *Catalyst 4500 Series Switch Module Installation Guide* for console interface cable connection procedures.

To access the switch through the console interface, perform this task:

	Command	Purpose
Step 1	Switch> <b>enable</b>	From the user EXEC prompt (>), enter <b>enable</b> to change to enable mode (also known as privileged mode or privileged EXEC mode).
Step 2	Password: <i>password</i> Switch#	At the password prompt, enter the system password. The prompt (#) appears, indicating that you have accessed the CLI in enabled mode.
Step 3	Switch# <b>quit</b>	When you are finished executing the task command, exit the session.

After accessing the switch through the EIA/TIA-232 interface, you see this display:

```
Press Return for Console prompt

Switch> enable
Password:< >
Switch#
```

## Accessing the CLI Through Telnet



### Note

Before you make a Telnet connection to the switch, you must set the IP address for the switch. See the [“Configuring Physical Layer 3 Interfaces”](#) section on page 34-12.

The switch supports up to eight simultaneous Telnet sessions. Telnet sessions disconnect automatically after remaining idle for the period specified by the **exec-timeout** command.

To make a Telnet connection to the switch, perform this task:

	Command	Purpose
Step 1	<code>telnet {hostname   ip_addr}</code>	From the remote host, enter the <b>telnet</b> command and the name or IP address of the switch you want to access.
Step 2	Password: <code>password</code> Switch#	At the prompt, enter the password for the CLI. If no password has been configured, press <b>Return</b> .
Step 3		Enter the necessary commands to complete your desired tasks.
Step 4	Switch# <code>quit</code>	When finished, exit the Telnet session.

This example shows how to open a Telnet session to the switch:

```

unix_host% telnet Switch_1
Trying 172.20.52.40...
Connected to 172.20.52.40.
Escape character is '^]'.
User Access Verification
Password:< >
Switch_1> enable
Password:
Switch_1#
  
```

## Performing Command-Line Processing

Switch commands are not case sensitive. You can abbreviate commands and parameters if the abbreviations contain enough letters to be different from any other currently available commands or parameters.

You can scroll through the last 20 commands stored in the history buffer and enter or edit a command at the prompt. [Table 2-1](#) lists the keyboard shortcuts for entering and editing switch commands.

**Table 2-1** Keyboard Shortcuts

Keystrokes	Result
Press <b>Ctrl-B</b> or press the <b>Left Arrow</b> key <sup>1</sup>	Moves the cursor back one character.
Press <b>Ctrl-F</b> or press the <b>Right Arrow</b> key <sup>1</sup>	Moves the cursor forward one character.
Press <b>Ctrl-A</b>	Moves the cursor to the beginning of the command line.
Press <b>Ctrl-E</b>	Moves the cursor to the end of the command line.
Press <b>Esc-B</b>	Moves the cursor back one word.
Press <b>Esc-F</b>	Moves the cursor forward one word.

1. The Arrow keys function only on ANSI-compatible terminals, such as VT100s.

## Performing History Substitution

The history buffer stores the last 20 command lines you entered. History substitution enables you to access these command lines without retyping them. [Table 2-2](#) lists the history substitution commands.

**Table 2-2** History Substitution Commands

Command	Purpose
<b>Ctrl-P</b> or the <b>Up Arrow</b> key <sup>1</sup>	Recalls commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall older commands successively.
<b>Ctrl-N</b> or the <b>Down Arrow</b> key <sup>1</sup>	Returns to more recent commands in the history buffer after commands have been recalled with <b>Ctrl-P</b> or the <b>Up Arrow</b> key. Repeat the key sequence to recall more recent commands.
Switch# <b>show history</b>	Lists the last several commands you have entered in EXEC mode.

1. The Arrow keys function only on ANSI-compatible terminals such as VT100s.

## About Cisco IOS Command Modes



### Note

For complete information about Cisco IOS command modes, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide* and the *Cisco IOS Configuration Fundamentals Command Reference* at the following URLs:

[http://www.cisco.com/en/US/docs/ios/12\\_2/configfun/configuration/guide/ffun\\_c.html](http://www.cisco.com/en/US/docs/ios/12_2/configfun/configuration/guide/ffun_c.html)

[http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf\\_book.html](http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html)

The Cisco IOS user interface has many different modes: user EXEC, privileged EXEC (enable), global configuration, interface, subinterface, and protocol-specific. The commands available to you depend on which mode you are in. To get a list of the commands in a given mode, enter a question mark (?) at the system prompt. See the “[Getting a List of Commands and Syntax](#)” section on [page 2-5](#) for more information.

When you start a session on the switch, you begin in user mode, also called user EXEC mode. Only a small subset of commands are available in EXEC mode. To have access to all commands, you must enter privileged EXEC mode, also called enable mode. To access the privileged EXEC mode, you must enter a password. When you are in the privileged EXEC mode, you can enter any EXEC command or access global configuration mode. Most EXEC commands are one-time commands, such as **show** commands, which display the current configuration status, and **clear** commands, which reset counters or interfaces. The EXEC commands are not saved when the switch is rebooted.

The configuration modes allow you to make changes to the running configuration. If you save the configuration, these commands are stored when you reboot the switch. You must start in global configuration mode. From global configuration mode, you can enter interface configuration mode, subinterface configuration mode, and a variety of protocol-specific modes.

You use a separate mode called ROMMON when the switch cannot boot up properly. For example, the switch might enter ROMMON mode if it does not find a valid system image when it is booting, or if its configuration file is corrupted. For more information, see the “[ROMMON Command-Line Interface](#)” section on page 2-7.

Table 2-3 lists and describes frequently used Cisco IOS modes.

**Table 2-3** Frequently Used Cisco IOS Command Modes

Mode	What You Use It For	How to Access	Prompt
User EXEC	To connect to remote devices, change terminal settings on a temporary basis, perform basic tests, and display system information.	Log in.	Switch>
Privileged EXEC (enable)	To set operating parameters. The privileged command set includes the commands in user EXEC mode, as well as the <b>configure</b> command. Use the <b>configure</b> command to access the other command modes.	From user EXEC mode, enter the <b>enable</b> command and the enable password (if a password has been configured).	Switch#
Global configuration	To configure features that affect the system as a whole, such as the system time or switch name.	From privileged EXEC mode, enter the <b>configure terminal</b> command.	Switch(config)#
Interface configuration	To enable or modify the operation of a 10-Gigabit Ethernet, Gigabit Ethernet, or Fast Ethernet interface with <b>interface</b> commands.	From global configuration mode, enter the <b>interface type location</b> command.	Switch(config-if)#
Console configuration	To configure the console interface; from the directly connected console or the virtual terminal; used with Telnet.	From global configuration mode, enter the <b>line console 0</b> command.	Switch(config-line)#

The Cisco IOS command interpreter, called the EXEC, interprets and runs the commands you enter. You can abbreviate commands and keywords by entering just enough characters to make the command unique from other commands. For example, you can abbreviate the **show** command to **sh** and the **configure terminal** command to **conf t**.

When you type **exit**, the switch backs out one level. To exit configuration mode completely and return to privileged EXEC mode, press **Ctrl-Z**.

## Getting a List of Commands and Syntax

In any command mode, you can get a list of available commands by entering a question mark (?).

```
Switch> ?
```

To obtain a list of commands that begin with a particular character sequence, enter those characters followed by the question mark (?). Do not include a space before the question mark. This form of help is called word help, because it completes a word for you.

To list keywords or arguments, enter a question mark in place of a keyword or argument. Include a space before the question mark. This form of help is called command syntax help, because it reminds you which keywords or arguments are applicable based on the command, keywords, and arguments you have already entered.

```
Switch# configure ?
memory          Configure from NV memory
network         Configure from a TFTP network host
overwrite-network Overwrite NV memory from TFTP network host
terminal        Configure from the terminal
<cr>
```

To redisplay a command you previously entered, press the **Up Arrow** key or **Ctrl-P**. You can continue to press the **Up Arrow** key to see the last 20 commands you entered.

**Tip**

If you are having trouble entering a command, check the system prompt and enter the question mark (?) for a list of available commands. You might be in the wrong command mode or using incorrect syntax.

Type **exit** to return to the previous mode. Press **Ctrl-Z** or enter the **end** command in any mode to immediately return to privileged EXEC mode.

## Virtual Console for Standby Supervisor Engine

Catalyst 4500 series switches can be configured with 2 supervisor engines to provide redundancy. When the switch is powered, one of the supervisor engines becomes active and remains active until a switchover occurs. The other supervisor engine remains in standby mode.

Each supervisor engine has its own console port. Access to the standby supervisor engine is possible only through the console port of the standby supervisor engine. You must connect to the standby console to access, monitor or debug the standby supervisor.

Virtual Console for Standby Supervisor Engine enables you to access the standby console from the active supervisor engine without requiring a physical connection to the standby console. It uses IPC over EOBC to communicate with the standby supervisor engine and thus emulate the standby console on the active supervisor engine. Only one active standby console session is active at any time.

The virtual console for standby supervisor engine enables users who are logged onto the active supervisor engine to remotely execute **show** commands on the standby supervisor engine and view the results on the active supervisor engine. Virtual console is available only from the active supervisor engine.

You can access the standby virtual console from the active supervisor engine with the **attach module**, **session module**, or **remote login** commands on the active supervisor engine. You must be in privilege EXEC mode (level 15) to run these commands to access the standby console.

Once you enter the standby virtual console, the terminal prompt automatically changes to *hostname-standby-console#*, where *hostname* is the configured name of the switch. The prompt is restored back to the original prompt when you exit the virtual console.

You exit the virtual console with the **exit** or **quit** commands. When the inactivity period of the terminal on the active supervisor engine where you logged in exceeds the configured idle time, you are automatically logged out of the terminal on the active supervisor engine. In this case, the virtual console session is also terminated. Virtual console session is also automatically terminated when the standby is rebooted. After the standby boots up, you need to create another virtual console session.

To log in to the standby supervisor engine using a virtual console, enter the following command:

```
Switch# session module 2
Connecting to standby virtual console
Type "exit" or "quit" to end this session
Switch-standby-console# exit
```

If the standby console is not enabled, the following message appears:

```
Switch-standby-console#
Standby console disabled.
Valid commands are: exit, logout
```

Virtual session into the standby console is N/A with RPR:

```
Switch# session module 2
IPC server port name IFConsoleServer:2 not registered on standby.
Secondary cannot be accessed by virtual console
```

**Note**

The standby virtual console provides the standard features that are available from the supervisor console such as command history, command completion, command help and partial command keywords.

The following limitations apply to the standby virtual console:

- All commands on the virtual console run to completion. It does not provide the auto-more feature; it behaves as if the **terminal length 0** command has been executed. It is also noninteractive. A executing command cannot be interrupted or aborted by any key sequence on the active supervisor engine. If a command produces considerable output, the virtual console displays it on the supervisor screen.
- The virtual console is noninteractive. Because the virtual console does not detect the interactive nature of a command, any command that requires user interaction causes the virtual console to wait until the RPC timer aborts the command.
- The virtual console timer is set to 60 seconds. The virtual console returns to its prompt after 60 seconds. During this time, you cannot abort the command from the keyboard. You must wait for the timer to expire before you continue.
- You cannot use virtual console to view debug and syslog messages that are being displayed on the standby supervisor engine. The virtual console only displays the output of commands that are executed from the virtual console. Other information that is displayed on the real standby console does not appear on the virtual console.

## ROMMON Command-Line Interface

ROMMON is a ROM-based program that is involved at power-up or reset, or when a fatal exception error occurs. The switch enters ROMMON mode if the switch does not find a valid software image, if the NVRAM configuration is corrupted, or if the configuration register is set to enter ROMMON mode. From the ROMMON mode, you can load a software image manually from flash memory, from a network server file, or from bootflash.

You can also enter ROMMON mode by restarting the switch and pressing **Ctrl-C** during the first five seconds of startup.

**Note**

**Ctrl-C** is always enabled for 60 seconds after you reboot the switch, even if **Ctrl-C** is configured to be off in the configuration register settings.

When you enter ROMMON mode, the prompt changes to **rommon 1>**. Use the **?** command to see the available ROMMON commands.

For more information about the ROMMON commands, refer to the command reference guide.

## Archiving Crashfiles Information

This feature allows you to archive crashinfo files (otherwise overwritten if another system reset were to happen first to the bootflash). Having access to archived crashinfo data greatly assists in troubleshooting.

To archive crashinfo files, perform this task:

	Command	Purpose
Step 1	Switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	Switch(config)# <b>exception crashinfo file bootflash: name</b>	Enables archiving crashinfo files to bootflash. The files are stored in bootflash with the <i>name</i> specified concatenated with the date.
Step 3	Switch(config)# <b>end</b>	Returns to privileged EXEC mode.
Step 4	Switch# <b>show running-config</b>	Verifies your entries.
Step 5	Switch# <b>copy running-config startup-config</b>	(Optional) Saves your entries in the configuration file.

## Displaying a Crash Dump for Supervisor Engine 6-E and 6L-E

A crash dump provides the following information:

- Malloc or free traces
- Chuck alloc/free traces
- Process block dump
- Register memory dump
- Current proc stack partial decode
- Interrupt level stack
- Last 128 memory block dump

To display a crash dump, do the following:

```
Switch# show platform crashdump

Current Time: 9/6/2010 15:47:21

Last Power Failure: 09/06/2010 15:03:28
Last Reload Status: 00002000
Last Software Reset State: 00000000

Crashdump version: 1

Last crash: 09/06/2010 06:21:58

Build: 12.2(20100723:074204) ENTSERVICES
buildversion addr: 14847D24
```



```

===== Context =====
pc=10999E70 lr=10999E34 msr=02029230 vector=00000600
cr=20004022 ctr=108EC3EC xer=00000000
r0=10999E34 r1=2421F930 r2=0000001E r3=234BBFD8
r4=0000000A r5=00000000 r6=2421F918 r7=00000000
r8=00000000 r9=00000000 r10=14850000 r11=234BBFD4
r12=EB93A100 r13=B4E9F3F3 r14=10CD0984 r15=00000000
r16=156CA504 r17=156CA504 r18=00000001 r19=00000000
r20=00000000 r21=00000000 r22=00000000 r23=00000000
r24=00000000 r25=00000000 r26=00000000 r27=00000000
r28=15870804 r29=00000000 r30=14850000 r31=00000000
dec=00083695 tbu=00000002 tbl=2A7D42DA
dar=80210020 dsisr=80210020 hid0=80004000

```

```

Traceback: 10999E70 11B430B8 10C84444 10C83338 11BE0C5C 10C93874 10C93D78 10C94140
10C992EC 10CD155C 1099BCFC 10992CEC

```

```

===== Stack frames =====
Frame 1: pc=11B430B8 stack=2421F940
Frame 2: pc=10C84444 stack=2421F948
Frame 3: pc=10C83338 stack=2421F9B0
Frame 4: pc=11BE0C5C stack=2421F9E8
Frame 5: pc=10C93874 stack=2421FA00
Frame 6: pc=10C93D78 stack=2421FA18
Frame 7: pc=10C94140 stack=2421FA48
Frame 8: pc=10C992EC stack=2421FA58
Frame 9: pc=10CD155C stack=2421FA70
Frame 10: pc=1099BCFC stack=2421FB08
Frame 11: pc=10992CEC stack=2421FB10

```

```

===== Pushed stack =====
2421F930: 2421F940 10999E34 2421F940 15868B74
2421F940: 2421F948 11B430B8 2421F9B0 10C84444
2421F950: 2421F978 00000000 00000000 00000000
2421F960: 00000000 2421F9C0 00000000 240CC3C8
2421F970: 2421F990 11AE7394 00000006 FFFFFFFF
2421F980: 00000000 00000000 00000000 14BE0000
2421F990: 00000000 00000000 00000000 00000000
2421F9A0: 00000001 00000000 15868B74 15868B74
2421F9B0: 2421F9E8 10C83338 00000000 00000000
2421F9C0: 00000071 15868B74 156CA328 13794ACD
2421F9D0: 00000000 00000001 00000000 1511A790
2421F9E0: 2366B680 15868B74 2421FA00 11BE0C5C
2421F9F0: 156CA328 156CA328 2366B680 15868B74
2421FA00: 2421FA18 10C93874 2421FA20 00000000
2421FA10: 00000000 2366B628 2421FA48 10C93D78
2421FA20: 2421FA58 10C95370 00000000 11BB0A98
2421FA30: 00000000 00000000 15868B74 00000000
2421FA40: 00000000 15868B74 2421FA58 10C94140
2421FA50: 00000003 15868B74 2421FA70 10C992EC
2421FA60: 00000000 00000000 00000000 156CA328
2421FA70: 2421FB08 10CD155C 0DFFFFFF FFFFFFFF
2421FA80: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
2421FA90: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
2421FAA0: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
2421FAB0: 00000001 FFFFFFFF FFFFFFFF FFFFFFFF
2421FAC0: FFFFFFFF 00000000 00000000 00000000
2421FAD0: 00000000 00000000 00000000 00000000
2421FAE0: 00000000 00000000 00000000 00000000

```

```

2421FAF0: 00000000 00000000 00000000 00000000
2421FB00: 00000000 00000000 2421FB10 1099BCFC
2421FB10: 00000000 10992CEC FFFFFFFF

```

```

===== Popped stack =====
2421F730: E8000800 151B1AB0 2421F748 132BBFA8
2421F740: 000E8000 151B1AB0 2421F760 132BC0D0
2421F750: 000E8000 00009B0A E8000800 151B1AB0
2421F760: 2421F778 132BC2A0 E8000800 00009B0A
2421F770: 00000800 153B1B7C 2421F790 123FAF28
2421F780: 2421F790 00000000 0000000A 151B17E4
2421F790: 2421F798 123FB2BC 2421F7B0 11C12A90
2421F7A0: 00009B0A 11C12880 0000000A 146C0000
2421F7B0: 2421F7C0 11BA7384 00000000 146B0000
2421F7C0: 2421F7D0 11AD3144 0000000A 0000000A
2421F7D0: 2421F7D8 11C10390 2421F7E0 11BB0424
2421F7E0: 2421F7F0 11BB04E4 2433FCD4 FFFFFFFE
2421F7F0: 2421F800 107CF880 7FFFFFFF FFFFFFFE
2421F800: 2421F8A8 107CCDF0 20637261 73686475
2421F810: 6D700000 00000000 2421F840 00000000
2421F820: 2421F8B0 00000000 0000004A 002E8A00
2421F830: 39760000 0000004A 00000000 2433FCF0
2421F840: 2421F848 2433FCF0 00000000 11A12ACC
2421F850: 13CD617C 10C7DAAC 00000000 2421F8AC
2421F860: 10CD0984 00000000 156CA504 156CA504
2421F870: 00000001 00000000 00000000 00000000
2421F880: 00000000 00000000 00000000 00000000
2421F890: 00000000 00000000 15870804 00000000
2421F8A0: 14850000 FFFFFFFE 2421F930 107CFC7C
2421F8B0: 2421F8C8 14BB1760 00000002 00000000
2421F8C0: 2421F930 14620E40 24330AB4 0000004A
2421F8D0: 00000000 00000000 2421F8E8 10C1FD9C
2421F8E0: 2421F8F8 00000000 00000000 00000000
2421F8F0: 15868B74 15868B74 2421F910 117CF5C0
2421F900: 2421F968 1586A45C 2421F920 15868B74
2421F910: 2421F918 00000000 14850000 00000000
2421F920: 2421F930 10999978 2421F930 00000000

```

```

===== Malloc and Free Traces=====

```

```

MallocFree Trace: ixmallocfree=0x2C ptr=0x151A40D8
151A3F78: 2366B628 11AF1144 2366B628 11AF1348 2366B66C 60000024 2447A940 11AF1350
151A3F98: 2447A940 30000018 2447A940 11AF1110 2366B628 4000000E 2366B628 11AF1144
151A3FB8: 2366B628 11AF1348 2366B66C 60000024 2447A940 11AF1350 2447A940 30000018
151A3FD8: 2447A940 11AF1110 2366B628 4000000E 2366B628 11AF1144 2366B628 11AF1348
151A3FF8: 2366B66C 60000024 2447A940 11AF1350 2447A940 30000018 2447A940 11AF1110
151A4018: 2366B628 4000000E 2366B628 11AF1144 2366B628 11AF1348 2366B66C 60000024
151A4038: 2447A940 11AF1350 22FAC944 119F6CC0 22FACF4C 6000005E 2433FCD4 40000046
151A4058: 2433FCD4 11A31DD4 2433FCD4 11A32370 2433FD88 6000005E 2447A940 30000018
151A4078: 2447A940 107D7294 2366B628 40000018 2366B628 10C9533C 2366B680 3000001A
151A4098: 2366B680 10C9536C 2433FCD4 4000000E 2433FCD4 10C7DAAC 2433FCD4 10C7DB90
151A40B8: 2433FD18 60000096 2433FCD4 4000000E 2433FCD4 10E28604 2433FCD4 10E287BC
151A40D8: 2433FD18 60000096 2366B66C 60000024 2447A940 11AF1350 2447A940 30000018
151A40F8: 2447A940 11AF1110 2366B628 4000000E 2366B628 11AF1144 2366B628 11AF1348
151A4118: 2366B66C 60000024 2447A940 11AF1350 2447A940 30000018 2447A940 11AF1110
151A4138: 2366B628 4000000E 2366B628 11AF1144 2366B628 11AF1348 2366B66C 60000024
151A4158: 2447A940 11AF1350 2447A940 30000018 2447A940 11AF1110 2366B628 4000000E

```

```

===== Chunk Malloc and Chunk Free Traces=====

```

```

151A3B78: 238928B8 11A32D70 11A34618 238928B8 11A3187C 11A34618
151A3B60: 15866F0C 10C7FF20 10C7F104 1586FBF0 10C7FE38 10C7F17C

```

151A3B48: 1586D760 10C7FE38 10C7F17C 1586FF98 10C7FE38 10C7F17C

```

151A3B30: 1586D760 10C84B24 10C7F17C 1586D760 10C7FE38 10C7F17C
151A3B18: 1586FF98 10C84B24 10C7F17C 1586FF98 10C7FE38 10C7F17C
151A3B00: 1586D760 10C84B24 10C7F17C 1586D760 10C7FE38 10C7F17C
151A3AE8: 1586FF98 10C84B24 10C7F17C 1586FF98 10C7FE38 10C7F17C
151A3AD0: 1586D760 10C84B24 10C7F17C 1586FBF0 10C84B24 10C7F17C
151A3AB8: 1586FBF0 10C7FE38 10C7F17C 1586D760 10C7FE38 10C7F17C
151A3AA0: 1586FBF0 10C84B24 10C7F17C 1586FBF0 10C7FE38 10C7F17C
151A3A88: 15870340 10C7FE38 10C7F17C 1586FBF0 10C84B24 10C7F17C
151A3A70: 1586D760 10C84B24 10C7F17C 1586D760 10C7FE38 10C7F17C
151A3A58: 1586FBF0 10C7FE38 10C7F17C 1586D760 10C84B24 10C7F17C
151A3A40: 1586D760 10C7FE38 10C7F17C 1586FBF0 10C84B24 10C7F17C
151A3A28: 1586FBF0 10C7FE38 10C7F17C 1586D760 10C84B24 10C7F17C
151A3A10: 15870340 10C84B24 10C7F17C 15870340 10C7FE38 10C7F17C
151A39F8: 1586D760 10C7FE38 10C7F17C 15870340 10C84B24 10C7F17C
151A39E0: 15870340 10C7FE38 10C7F17C 1586D760 10C84B24 10C7F17C
151A39C8: 1586D760 10C7FE38 10C7F17C 15870340 10C84B24 10C7F17C
151A39B0: 15870340 10C7FE38 10C7F17C 1586D760 10C84B24 10C7F17C
151A3998: 1586D760 10C7FE38 10C7F17C 15870340 10C84B24 10C7F17C
151A3980: 15870340 10C7FE38 10C7F17C 1586D760 10C84B24 10C7F17C
151A3968: 1586D760 10C7FE38 10C7F17C 15870340 10C84B24 10C7F17C
151A3950: 15870340 10C7FE38 10C7F17C 1586D3B8 10C7FE38 10C7F17C
151A3938: 15870340 10C84B24 10C7F17C 1586D760 10C84B24 10C7F17C
151A3920: 1586D760 10C7FE38 10C7F17C
151A3C14: 15870340 10C7FE38 10C7F17C 1586D760 10C84B24 10C7F17C
151A3BFC: 1586D3B8 10C84B24 10C7F17C 1586D3B8 10C7FE38 10C7F17C
151A3BE4: 1586D760 10C7FE38 10C7F17C 1586D3B8 10C84B24 10C7F17C
151A3BCC: 15870340 10C84B24 10C7F17C 15870340 10C7FE38 10C7F17C
151A3BB4: 1586D3B8 10C7FE38 10C7F17C 15870340 10C84B24 10C7F17C
151A3B9C: 1586D760 10C84B24 10C7F17C 1586D760 10C7FE38 10C7F17C
151A3B84: 15870340 10C7FE38 10C7F17C

```

=====  
Process Level Info  
=====

---- Current Process Block (at 0x24330AB4) ----

```

24330A8C: AB1234CD 710000 24330AB4 13DF55F0 11A2F280 24330D48 24330A5C 8000014A
24330AAC: 1 10530DC4 242110BC 1582AAAC 156CA328 10CD0984 0 156CA504
24330ACC: 156CA504 6 FFFFFFFF 1 2421FA78 13D2A3E0 FF 0
24330AEC: 1 13D2A3E0 2421FA78 24330AB4 14BE0000 156CA328 107D4240 40004024
24330B0C: 11A3C6A4 2029230 0 0 0 0 0 10100
24330B2C: 0 1000000 0 0 0 71 0 0
24330B4C: 0 25610 2350 320BC 0 0 0 0
24330B6C: 0 2035F 0 156CA328 0 2210B 0 2210B
24330B8C: 0 13 0 13D42FC4 4 1 15E 1
24330BAC: 0 0 0 EA60 EA60 156CA328 0 0
24330BCC: 0 0 0 0 0 24330AB4 151A5708 0
24330BEC: 0 0 149A2408 0 0 0 420A 0
24330C0C: 0 0 24330BEC 0 0 0 4290 24330AB4
24330C2C: 0 0 0 0 24330BEC 24330AB4 0 0
24330C4C: 142D2 0 0 0 24330BEC 24330AB4 0 0
24330C6C: 242D2 0 0 0 0 0 0 24325EB4
24330C8C: 0 0 0 0 0 0 0 0
24330CAC: 0 24325F14 24330C9C 24325EBC 0 151A6450 0 0
24330CCC: 0 0 0 FFFFFFFF FFFFFFFF 0 0 0
24330CEC: 0 0 0 0 0 23EFC15C 0 0
24330D0C: 32 0 0 0 0 0 0 0
24330D2C: 0 0 0 0 BEEFCAFE 0

```

---- Partial decode of process block ----

Pid 113: Process "Exec" stack 0x242110BC savedsp 0x1582AAAC

```

Flags: analyze crashblock on_old_queue
Status      0x00000000 Orig_ra  0x00000000 Routine    0x00000000 Signal  0
Caller_pc   0x00000000 Callee_pc 0x00000000 Dbg_events 0x00000000 State   0
Totmalloc   153104      Totfree   9040      Totgetbuf  0
Totretbuf   0          Edisms    0x0       Eparm      0x156CA328
Elapsed     0x0        Ncalls    0x13      Ngiveups   0x0
Priority_q   4          Ticks_5s  1         Cpu_5sec   0          Cpu_1min  0
Cpu_smin    0          Stacksize 0xEA60    Lowstack   0xEA60
Ttyptr      0x156CA328 Mem_holding 0x320BC   Thrash_count 0
Wakeup_reasons 0x0FFFFFFF Default_wakeup_reasons 0x0FFFFFFF
Direct_wakeup_major 0x00000000 Direct_wakeup_minor 0x00000000

```

```

Regs R14-R31, CR, PC, MSR at last suspend; R3 from proc creation, PC unused:
R3 : 156CA328 R14: 10CD0984 R15: 00000000 R16: 156CA504 R17: 156CA504
R18: 00000006 R19: FFFFFFFF R20: 00000001 R21: 2421FA78 R22: 13D2A3E0
R23: 000000FF R24: 00000000 R25: 00000001 R26: 13D2A3E0 R27: 2421FA78
R28: 24330AB4 R29: 14BE0000 R30: 156CA328 R31: 107D4240 CR: 40004024
PC : 11A3C6A4 MSR: 02029230

```

---- Current Process Stack (0x714 bytes used, out of 0xEA60 available) ----

Current SP = 0x2421F930, saved SP = 0x1582AAAC

```

2421F71C:      1A 2421F918      0 FFFFFFFF 151B1AB0 E8000800 151B1AB0 2421F748
2421F73C:     132BBFA8      E8000 151B1AB0 2421F760 132BC0D0      E8000      9B0A E8000800
2421F75C:     151B1AB0 2421F778 132BC2A0 E8000800      9B0A      800 153B1B7C 2421F790
2421F77C:     123FAF28 2421F790      0          A 151B17E4 2421F798 123FB2BC 2421F7B0
2421F79C:     11C12A90      9B0A 11C12880      A 146C0000 2421F7C0 11BA7384      0
2421F7BC:     146B0000 2421F7D0 11AD3144      A          A 2421F7D8 11C10390 2421F7E0
2421F7DC:     11BB0424 2421F7F0 11BB04E4 2433FCD4 FFFFFFFE 2421F800 107CF880 7FFFFFFF
2421F7FC:     FFFFFFFE 2421F8A8 107CCDF0 20637261 73686475 6D700000      0 2421F840
2421F81C:      0 2421F8B0      0          4A 2E8A00 39760000      4A      0
2421F83C:     2433FCF0 2421F848 2433FCF0      0 11A12ACC 13CD617C 10C7DAAC      0
2421F85C:     2421F8AC 10CD0984      0 156CA504 156CA504      1      0      0
2421F87C:      0      0      0      0      0      0      0 15870804
2421F89C:      0 14850000 FFFFFFFE 2421F930 107CFC7C 2421F8C8 14BB1760      2
2421F8BC:      0 2421F930 14620E40 24330AB4      4A      0      0 2421F8E8
2421F8DC:     10C1FD9C 2421F8F8      0      0      0 15868B74 15868B74 2421F910
2421F8FC:     117CF5C0 2421F968 1586A45C 2421F920 15868B74 2421F918      0 14850000
2421F91C:      0 2421F930 10999978 2421F930      0 2421F940 10999E34 2421F940
2421F93C:     15868B74 2421F948 11B430B8 2421F9B0 10C84444 2421F978      0      0
2421F95C:      0      0 2421F9C0      0 240CC3C8 2421F990 11AE7394      6
2421F97C:     FFFFFFFF      0      0      0 14BE0000      0      0      0
2421F99C:      0      1      0 15868B74 15868B74 2421F9E8 10C83338      0
2421F9BC:      0      71 15868B74 156CA328 13794ACD      0      1      0
2421F9DC:     1511A790 2366B680 15868B74 2421FA00 11BE0C5C 156CA328 156CA328 2366B680
2421F9FC:     15868B74 2421FA18 10C93874 2421FA20      0      0 2366B628 2421FA48
2421FA1C:     10C93D78 2421FA58 10C95370      0 11BB0A98      0      0 15868B74
2421FA3C:      0      0 15868B74 2421FA58 10C94140      3 15868B74 2421FA70
2421FA5C:     10C992EC      0      0      0 156CA328 2421FB08 10CD155C DFFFFFFF
2421FA7C:     FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
2421FA9C:     FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF      1 FFFFFFFF FFFFFFFF
2421FABC:     FFFFFFFF FFFFFFFF      0      0      0      0      0      0
2421FADC:      0      0      0      0      0      0      0      0
2421FAFC:      0      0      0 2421FB10 1099BCFC      0 10992CEC FFFFFFFF

```

===== Interrupt Level Stack Dump =====

---- Level 1 Interrupt Stack (0x0 bytes used, out of 0x2328 available) ----

```

intstacks[1]: base 0x156DB3D8 stack 0x156DD6F8 routine 0x0      count 0x0
                size 0x2328      low 0x2328      desc 0x156BE7D0

```

```

---- Level 2 Interrupt stack (0x3F8 bytes used, out of 0x2328 available) ----

intstacks[2]: base 0x156D90B0 stack 0x156DB3D0 routine 0x0 count 0x2
              size 0x2328 low 0x2328 desc 0x156C0C78

156DAFE0: 156DAFE8 FFFFFFFF 156DB020 119E1374 0 B6B8 0 B6F4
156DB000: 156DB020 16035650 156DB0E0 0 2DAE 4 1 16031964
156DB020: 156DB028 119E15BC 156DB050 119E1670 0 B6B8 0 1E3
156DB040: 2DAE 1603191C 156DB050 1603190C 156DB0D0 11BB458C FFFFFFFF FFFFFFFF
156DB060: 0 1E3 16002438 1603191C 1CCB58E0 64 0 2DAE
156DB080: 0 B6B8 FFFFFFFF FFFFFFFF FFFFFFFF 137B49A8 1603560C 160355D0
156DB0A0: 14BABC00 B9DE8DC0 156DB128 0 1C703D84 17F1C788 0 11
156DB0C0: 0 156DB138 160355D0 156DB128 156DB100 11EBBCDC 0 11
156DB0E0: 1CCB58E0 64 156DB110 2DAE 14BAC400 156DB220 17B7B610 0
156DB100: 90040008 151B1AB0 0 151B1AB0 156DB120 132BBFA8 90040 122C3E40
156DB120: 156DB138 132BC0D0 1CCB58E0 0 0 151B1AB0 156DB150 132BCC08
156DB140: 0 156DB240 156DB158 10 156DB1C0 15 156DB170 129ADCC4
156DB160: 156DB170 11 156DB1E0 16 156DB3B0 122BF51C FFFFFFFF FFFFFFFF
156DB180: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
156DB1A0: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
156DB1C0: 0 2000000 0 7FFFFFFF 0 0 FFFFFFFF FFFFFFFF
156DB1E0: 0 20001FF 0 7FFFFFFF 0 0 FFFFFFFF FFFFFFFF
156DB200: 10100 1F4 1F4 77359400 3 2 16 3D
156DB220: 294 294 294 0 0 2 2 1
156DB240: 80000 0 0 FF 0 0 FFFFFFFF FFFFFFFF
156DB260: 0 1EB 0 1FF 0 0 FFFFFFFF FFFFFFFF
156DB280: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
156DB2A0: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
156DB2C0: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
156DB2E0: 0 FFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
156DB300: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
156DB320: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
156DB340: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
156DB360: 15FFFFFF 10 10 10 10 FFFFFFFF FFFFFFFF FFFFFFFF
156DB380: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 0 14BE0000 146CF310
156DB3A0: 146D0000 14620E80 4 1C7061FC 156DB3C0 132DE01C FFFFFFFF 2
156DB3C0: 156DB3C8 132DDD84 14BAE470 11C0DCD0 FFFFFFFF FFFFFFFF

---- Level 3 Interrupt stack (0x350 bytes used, out of 0x2328 available) ----

intstacks[3]: base 0x156D6D88 stack 0x156D90A8 routine 0x0 count 0x995
              size 0x2328 low 0x2328 desc 0x156BE924

156D8D60: 156D8D68 FFFFFFFF 156D8DA0 119E1374 0 21EEB 0 21EE4
156D8D80: 0 1 1 160BA724 0 0 156D8DA0 160BA85C
156D8DA0: 156D8DD8 119E1E40 156D8DB0 1 1603560C 156D8F38 160029D8 16035650
156D8DC0: 1 4 0 22030 156D8DD8 160A5670 156D8DF8 119E1F74
156D8DE0: 156D8DF8 4 156D8E08 160BA85C 156D8E08 4 4 160BA85C
156D8E00: 156D8E08 160BA724 156D8E38 11A312A8 156D8E30 119DF688 156D8E30 16035650
156D8E20: 0 16002CA8 156D8E50 16002F78 156D8E50 119DFBD8 0 8E12
156D8E40: 0 156D8F38 156D8E50 1603148C 156D8ED0 11BB458C 156D8E60 1603148C
156D8E60: 0 8E12 16002CA8 156D8F38 1C6FF080 64 0 0
156D8E80: 156D8E90 156D8EE0 156D8E90 1C6FEF9C 156D8ED0 13B40000 14370000 14BC97D0
156D8EA0: 156D8F28 2980A1B9 156D8ED0 84D7317 0 0 1C6FEFAC 0
156D8EC0: 84D7317 1C6FEED4 0 153B1DA4 156D8ED8 11EBBCDC 156D8EE0 11C0C254
156D8EE0: 156D8EF8 132F05E0 2 2980A460 156D8EF8 84D7317 156D8F00 132EFB10
156D8F00: 156D8F18 132B19FC 84D7317 1C6FEED4 0 84D7317 156D8F90 132B1EEC
156D8F20: 0 84D6D76 1C6FF080 64 0 0 0 8E12
156D8F40: 16002CA8 156D8F38 14BE0000 13FD0000 138A0000 160BA4B0 160BA4B0 160BA4B0
156D8F60: 14BABC00 2980A1B9 137C0000 13F50000 14BAC400 0 14BE0000 0
156D8F80: 84D7317 1C6FEED4 0 84D7317 156D8FA8 132B2448 156D8FA0 156D8FB8
156D8FA0: 156D8FA8 11BBE798 156D9030 132B0C9C 156D8FB8 11BBE798 0 7530
156D8FC0: 0 2EE0 0 1 FFFFFFFF FFFFFFFF 4B354370 754D616E

```

```

156D8FE0: 20526576 69657700      0      0      0      0      0      0
156D9000:      0      0 1ADBEEF 1896AD90 156D9030      0      0 146CF310
156D9020: 146D0000 14620EA0      D 1893E4BC 156D9038 134D23A4 156D9058 12023A6C
156D9040:      0 1B1DDC40 156D9050      40      D 1B1DDC40 156D9080 11ED3534
156D9060:      40 132D6244      0 14620EA0 146D0000 14620EA0      D 22D85610
156D9080: 156D9088 133C43C8 156D9098 132DE01C FFFFFFFF      3 156D90A0 132DDE4C
156D90A0: 14BAE470 11C0DCD0 FFFFFFFF FFFFFFFF

```

---- Level 4 Interrupt stack (0x348 bytes used, out of 0x2328 available) ----

```

intstacks[4]: base 0x156D4A60 stack 0x156D6D80 routine 0x0      count 0x8376
              size 0x2328      low 0x2328      desc 0x156BEA78

156D6A40: 156D6A48 FFFFFFFF 156D6A80 119E1374      0 21E4B      0 21E48
156D6A60: FFFFFFFF      1      1 160BA724      0      0 156D6A80 160BA85C
156D6A80: 156D6AB8 119E1E40 FFFFFFFF      1 1603560C 156D6C18 16002938 16035650
156D6AA0:      1      4      0 21F90 156D6AB8 160A5670 156D6AD8 119E1F74
156D6AC0: 156D6AD8      4      4 160BA85C 156D6AD8 160BA724 156D6B08 11A312A8
156D6AE0: 156D6B00 119DF688 156D6BC0 16035650      0 156C504C      1 160BA724
156D6B00: 156D6B08 156C8B5C 156D6B30 11A31B54 156D6B30 119DFBD8      0 8DCE
156D6B20:      0 1603129C 156D6B30 1603128C 156D6BB0 11BB458C FFFFFFFF FFFFFFFF
156D6B40:      0 8DCE 160028E8 1603129C 1BB124AC      64      0      0
156D6B60:      0 1A68C FFFFFFFF FFFFFFFF FFFFFFFF 13B40000 14370000 14BC97D0
156D6B80: 156D6C08 28E47C74 156D6BB0 84B1B7D      0      0 1BB123D8      0
156D6BA0: 84B1B7D 1BB12300      0 153B1DA4 156D6BB8 11EBBCDC 156D6BC0 11C0C254
156D6BC0: 156D6BD8 132F05E0      2 28E47EC0 156D6BD8 84B1B7D 156D6BE0 132EFB10
156D6BE0: 156D6BF8 132B19FC 84B1B7D 1BB12300      0 84B1B7D 156D6C70 132B1EEC
156D6C00: FFFFFFFF FFFFFFFF 1BB124AC      64      0      0      0 8DCE
156D6C20: 160028E8 1603129C 14BE0000 160BA4D8 13860000 13FA0000 160BA428 160BA4B0
156D6C40: 14BABC00 28E47C74 137C0000 13F50000 14BAC400      0 14BE0000      0
156D6C60: 84B1B7D 1BB12300      0 84B1B7D 156D6C88 132B2448 156D6C80 156D6C98
156D6C80: 156D6C88 11BBE798 156D6D10 132B0C9C A0000 14800000      0 1770
156D6CA0:      0 BB8 156D6CD0      1 14BAC400      0 14BE0000 146CF310
156D6CC0: 146D0000 151B1AB0 156D6CF0 151B1AB0 156D6CE0      0 156D6D10 146CF310
156D6CE0: 2029230 14620EC0 156D6D20 153B1DA4 156D6CF8 151B1AB0 3012000 11C0C254
156D6D00: 156D6D18 132F05E0 3012020 153B1C8C 156D6D30 11C0FE70 156D6D20 153B1C8C
156D6D20: 156D6D40 11C0FE28 156D6D38 153B1C8C 3012040 153B1C8C 156D6D48 11C100CC
156D6D40: 156D6D50 11C10348 14BE0000 146C62B4 156D6D78 11BB0A10 FFFFFF 1CCAFCB8
156D6D60: 156D6D70 146CF310 146D0000 14620EC0      2A      4 14BAE470 11C0DCD0
156D6D80: FFFFFFFF FFFFFFFF

```

---- Level 5 Interrupt stack (0x170 bytes used, out of 0x2328 available) ----

```

intstacks[5]: base 0x156D2738 stack 0x156D4A58 routine 0x0      count 0x8843
              size 0x2328      low 0x2328      desc 0x156BEBCC

156D48F0: 156D4918 FFFFFFFF 14BAC400      0 14BE0000 13B50000      0 151B1AB0
156D4910: A0000060 151B1AB0 156D4928 132BBFA8 A0000 151B1AB0 156D4940 132BC0D0
156D4930: A0000 14C00000 A0000060 151B1AB0 156D4958 132BC42C A0000060 14C00000
156D4950:      60 189A84E0 156D4970 12405FD8 156D4970      1      1 1B1AD9E0
156D4970: 156D4978 124067F8 156D4998 11CB7020 156D49A8 14C00000 FFFFFFFF      0
156D4990: 14BE0000 13B50000      2 1B5A1068 2029230 153B1DA4 156D49D0 1BB11BE0
156D49B0: 156D49B8      0 14BE0000 13B50000      3 14380000 1B1AD9E0 153B1DA4
156D49D0: 156D49D8 132EFB10 156D49E0 11C0C254 156D49F8 132F05E0      2 2A7A9FE0
156D49F0: 2029230 153B1DA4 156D4A00 132EFB10 156D4A18 11CB7200      3 14380000
156D4A10: 14380000 153B1DA4 156D4A20 1338A684 156D4A48 132F0B04      2 2A7A9F6F
156D4A30: FFFFFFFF 146CF310 146D0000 14620EE0      0      5 156D4A50 11BAE8C0
156D4A50: 14BAE470 11C0DCD0 FFFFFFFF FFFFFFFF

```

---- Level 6 Interrupt stack (0x0 bytes used, out of 0x2328 available) ----

```

intstacks[6]: base 0x156D0410 stack 0x156D2730 routine 0x0      count 0x0
              size 0x2328      low 0x2328      desc 0x156BED20

```

```

---- Level 7 Interrupt stack (0x0 bytes used, out of 0x2328 available) ----
intstacks[7]: base 0x156CE0E8 stack 0x156D0408 routine 0x0      count 0x0
              size 0x2328      low  0x2328      desc   0x156BEE74

---- Level 8 Interrupt stack (base 0x0, size 0x0) is invalid ----

---- Level 9 Interrupt stack (base 0x0, size 0x0) is invalid ----

===== Register Memory Dump =====

Reg00(PC ): 10999E70
Reg01(MSR): 2029230 [Not RAM Addr]
Reg02(CR ): 20004022
Reg03(LR ): 10999E34
Reg04(CTR): 108EC3EC
Reg05(XER): 0 [Not RAM Addr]
Reg06(DAR): 0 [Not RAM Addr]
Reg07(DSISR): 0 [Not RAM Addr]
Reg08(DEC): 83695 [Not RAM Addr]
Reg09(TBU): 2 [Not RAM Addr]
Reg10(TBL): 2A7D42DA
Reg11(IMMR): 0 [Not RAM Addr]
Reg12(R0 ): 10999E34
Reg13(R1 ): 2421F930
Reg14(R2 ): 1E [Not RAM Addr]
Reg15(R3 ): 234BBFD8 [In malloc Block 0x234BBB54] [Last malloc Block 0x234BBB10]
Reg16(R4 ): A [Not RAM Addr]
Reg17(R5 ): 0 [Not RAM Addr]
Reg18(R6 ): 2421F918
Reg19(R7 ): 0 [Not RAM Addr]
Reg20(R8 ): 0 [Not RAM Addr]
Reg21(R9 ): 0 [Not RAM Addr]
Reg22(R10): 14850000
Reg23(R11): 234BBFD4
Reg24(R12): EB93A100 [Not RAM Addr]
Reg25(R13): B4E9F3F3 [Not RAM Addr]
Reg26(R14): 10CD0984
Reg27(R15): 0 [Not RAM Addr]
Reg28(R16): 156CA504 [In malloc Block 0x156CA2F0]
Reg29(R17): 156CA504
Reg30(R18): 1 [Not RAM Addr]
Reg31(R19): 0 [Not RAM Addr]
Reg32(R20): 0 [Not RAM Addr]
Reg33(R21): 0 [Not RAM Addr]
Reg34(R22): 0 [Not RAM Addr]
Reg35(R23): 0 [Not RAM Addr]
Reg36(R24): 0 [Not RAM Addr]
Reg37(R25): 0 [Not RAM Addr]
Reg38(R26): 0 [Not RAM Addr]
Reg39(R27): 0 [Not RAM Addr]
Reg40(R28): 15870804 [In malloc Block 0x158707DC] [Last malloc Block 0x15870790]
Reg41(R29): 0 [Not RAM Addr]
Reg42(R30): 14850000
Reg43(R31): 0 [Not RAM Addr]

buffer check=0 sched_hc=0x0

---- block0 ptr=2421F8D0 is_malloc=0 length=0x260 ----

2421F890: 0 0 15870804 0 14850000 FFFFFFFE 2421F930 107CFC7C
2421F8B0: 2421F8C8 14BB1760 2 0 2421F930 14620E40 24330AB4 4A

```

```

2421F8D0:      0      0 2421F8E8 10C1FD9C 2421F8F8      0      0      0
2421F8F0: 15868B74 15868B74 2421F910 117CF5C0 2421F968 1586A45C 2421F920 15868B74
2421F910: 2421F918      0 14850000      0 2421F930 10999978 2421F930      0
2421F930: 2421F940 10999E34 2421F940 15868B74 2421F948 11B430B8 2421F9B0 10C84444
2421F950: 2421F978      0      0      0      0 2421F9C0      0 240CC3C8
2421F970: 2421F990 11AE7394      6 FFFFFFFF      0      0      0 14BE0000
2421F990:      0      0      0      0      1      0 15868B74 15868B74
2421F9B0: 2421F9E8 10C83338      0      0      71 15868B74 156CA328 13794ACD
2421F9D0:      0      1      0 1511A790 2366B680 15868B74 2421FA00 11BE0C5C
2421F9F0: 156CA328 156CA328 2366B680 15868B74 2421FA18 10C93874 2421FA20      0
2421FA10:      0 2366B628 2421FA48 10C93D78 2421FA58 10C95370      0 11BB0A98
2421FA30:      0      0 15868B74      0      0 15868B74 2421FA58 10C94140
2421FA50:      3 15868B74 2421FA70 10C992EC      0      0      0 156CA328
2421FA70: 2421FB08 10CD155C DFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
2421FA90: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
2421FAB0:      1 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF      0      0      0
2421FAD0:      0      0      0      0      0      0      0      0
2421FAF0:      0      0      0      0      0      0 2421FB10 1099BCFC
2421FB10:      0 10992CEC FFFFFFFF FD0110DF AB1234CD      0      0 1378AC50

```

```
---- block1 ptr=10999E34 is_malloc=0 length=0x13C ----
```

```

10999DF4: 3D2014BE 8009845C 2F800000 419E0010 3D2014BE 806983C4 4800000C 3D201485
10999E14: 80698C20 4927268D 38000000 3D2014BE 90098484 3D201485 90098CAC 4BFFFAB9
10999E34: 2F9F0000 419E0038 3D2014BE 80098498 2F800000 40BE0028 3860FFFE 3C8013CD
10999E54: 3884617C 4BE35D99 2F9F0003 409D0008 3BE00003 1C7F03E8 4909658D 7FE10808
10999E74: 38000000 3D201445 900900E4 80010014 7C0803A6 83E1000C 38210010 4E800020
10999E94: 9421FFF8 7C0802A6 9001000C 7C862378 38832010 38602000 3CA013CD 38A56198
10999EB4: 482E2201 8001000C 7C0803A6 38210008 4E800020 7C681B78 7C6A1B78 39200000
10999ED4: 89630000 2F8B0000 419E0078 380BFF9F 2B800005 40BD0010 380BFFD0 2B800009
10999EF4: 419D0060 552B2036 880A0000 7C000774 2F800039 419D0014 7D2B0214 3929FFD0
10999F14: 394A0001 48000018 892A0000 7D290774 7D2B4A14 3929FFA9 394A0001 896A0000
10999F34: 2F8B0000 419E001C 380BFF9F 2B800005 40BDFFB4 380BFFD0 2B800009 409DFFA8
10999F54: 38600000 7F8A4000 4D9E0020 91250000 91440000 38600001 4E800020

```

```
---- block2 ptr=20004020 is_malloc=0 length=0x100 ----
```

```

20003FE0:  ADBEEF      0      0      0      BEEF      0 7ADBEEF 1F724E78
20004000:  12800      0 1ADBEEF 12216BE8 1BE7FE60      BEEF 1BE7FE70 1813277C
20004020: 1BE7FE78 1253BA98      FE70 FFFFFFFF      3      0 FFFFFFFF FFFFFFFF
20004040: FFFFFFFF FFFFFFFF 10800      0      0 1FFFF 1BE7FF38 11F86C4C
20004060: 180AB988 1BE7FEF8 2027FEE8 FF84BEEF      3      0      0      0
20004080:      BEEF      0      BEEF      0      0      0      0      0
200040A0:      0  ADBEEF      0      0      0      BEEF      0 7ADBEEF
200040C0: 1F724E78 12800      0 1ADBEEF 12216BE8 1BE7FE60      BEEF 1BE7FE70
200040E0: 1813277C 1BE7FE78 1253BA98      FE70 FFFFFFFF      3      0 FFFFFFFF
20004100: FFFFFFFF FFFFFFFF FFFFFFFF 10800      0      0 1FFFF 1BE7FF38

```

```
---- block3 ptr=108EC3EC is_malloc=0 length=0x100 ----
```

```

108EC3AC: 7C0803A6 83E1000C 38210010 4E800020 9421FFF8 7C0802A6 9001000C 2C030000
108EC3CC: 40A2000C 4BFFF7F9 48000008 4BFFF79 8001000C 7C0803A6 38210008 4E800020
108EC3EC: 9421FFF8 7C0802A6 9001000C 3D20149A 8009FB08 2F800000 41BE0028 4BFEF791
108EC40C: 3D201485 81299C1C 81290024 7D2903A6 38600000 4E800421 38600000 4BFF0EFD
108EC42C: 8001000C 7C0803A6 38210008 4E800020 9421FFF8 7C0802A6 9001000C 3D20149A
108EC44C: 8009FB08 2F800000 41BE0028 4BFEF741 3D201485 81299C1C 81290024 7D2903A6
108EC46C: 38600000 4E800421 38600000 4BFF0EAD 8001000C 7C0803A6 38210008 4E800020
108EC48C: 9421FFE8 7C0802A6 BF810008 9001001C 7C9C2378 38000000 7C7E1B79 418200B0
108EC4AC: 7FC3F378 492150A5 38000000 7C7D1B79 4182009C 3BE00000 7F9FE800 40BC008C
108EC4CC: 7C9EF8AE 3D201442 8069EDA4 7C840774 49214EFD 2F830000 409E0064 2F9C0000

```

```
---- block4 ptr=234BBB10 is_malloc=1 length=0x100 ----
```

```
234BBAD0: FFFE0000      0 13C9C0B0 107FD290 234BBB10 234BBA24 8000000E      1
```



```

234BBAF0:      0 23056294 23054D90 13597D4C      1      0      0 FD0110DF
234BBB10: AB1234CD FFFE0000      0 13D9A594 10027870 234BBB54 234BBAE0 8000000E
234BBB30:      1      0      1      4      0      0      0      0
234BBB50: FD0110DF AB1234CD FFFE0000      0 156CD7F4 119EB018 234BC350 234BBB24
234BBB70: 800003EA      1 119F6768      0 234466EC 234FFE84      0 156CD7B8
234BBB90:      64      77      C000C      0      0      0      0      0
234BBB0:      30000 14BB1760 52656720 46756E63 74696F6E 20310000 234BBDB8 234BC34C
234BBBD0:      0      0      0 234BBDB8 234BBDC4 234BBDD0 234BBDDC 234BBDE8
234BBBF0: 234BBDF4 234BBE00 234BBE0C 234BBE18 234BBE24 234BBE30 234BBE3C 234BBE48

```

```
---- block5 ptr=15870790 is_malloc=1 length=0x14C ----
```

```

15870750:      0      0      0      0      0      0      0      0
15870770:      0      0      0      0      0      0      0 FD0110DF
15870790: AB1234CD FFFE0000      0 13D9A594 10CA1538 158707DC 1586B958 80000012
158707B0:      1 4928F581      0      1 23C0BED0      0      0      0
158707D0: 1449E540      F FD0110DF AB1234CD FFFE0000      0 13D2B910 10C89680
158707F0: 15870840 158707A4 8000001E      1 38210008 158711F0 13D40DB8      0
15870810: 13D3EC78      0      0      0 10CC65BC      7 144B0254 15870868
15870830: 158708AC      0      0 FD0110DF AB1234CD FFFE0000      0 13D2B91C
15870850: 10C896E8 15870884 158707F0 8000000E      1 7C09002E 158708F0 23F0FB18
15870870:      17      0      0      0 FD0110DF AB1234CD FFFE0000      0
15870890: 13D2B92C 10C8970C 158708C8 15870854 8000000E      1 7D6B4A14 15870980
158708B0: 15871160      8      0      0      0 FD0110DF AB1234CD FFFE0000
158708D0:      0 13D2BA48 10C8BE78

```

```
---- block6 ptr=15870790 is_malloc=1 length=0x100 ----
```

```

15870750:      0      0      0      0      0      0      0      0
15870770:      0      0      0      0      0      0      0 FD0110DF
15870790: AB1234CD FFFE0000      0 13D9A594 10CA1538 158707DC 1586B958 80000012
158707B0:      1 4928F581      0      1 23C0BED0      0      0      0
158707D0: 1449E540      F FD0110DF AB1234CD FFFE0000      0 13D2B910 10C89680
158707F0: 15870840 158707A4 8000001E      1 38210008 158711F0 13D40DB8      0
15870810: 13D3EC78      0      0      0 10CC65BC      7 144B0254 15870868
15870830: 158708AC      0      0 FD0110DF AB1234CD FFFE0000      0 13D2B91C
15870850: 10C896E8 15870884 158707F0 8000000E      1 7C09002E 158708F0 23F0FB18
15870870:      17      0      0      0 FD0110DF AB1234CD FFFE0000      0

```

```
---- block7 ptr=240CC354 is_malloc=1 length=0x14C ----
```

```

240CC314:      2 240CC37C      0      0      0      0      2      0
240CC334:      0      0      0      0      0      0      0 FD0110DF
240CC354: AB1234CD CD0000 24031228 240CC1F0 1011D6CC 240CC3A0 240CC2F0 80000012
240CC374:      1 D0D0D0D      1      8 10B4FEE8      0      2      4
240CC394:      0 10B50054 FD0110DF AB1234CD FFFE0000      0 13D9A594 10C8D690
240CC3B4: 240CC404 240CC368 8000001E      1 10C7DB80      1      40      0
240CC3D4:      1 23553660 10C8D61C      0      0      0      0      0
240CC3F4:      0      0      0 FD0110DF AB1234CD CD0000 24031228 1362B664
240CC414: 10DF2B24 240CEA20 240CC3B4 800012FA      1 D0D0D0D 2416C8DC ABADCAFE
240CC434:      C      C      0      CD 80000000      0      0      0
240CC454:      0      0      0      0      0      0      0      0
240CC474:      0      0      0      0      0      0      0      0
240CC494:      0      0      0

```

```
---- block8 ptr=13794ACC is_malloc=0 length=0x100 ----
```

```

13794A8C: 51522E2E 2E2E2E2E 5C2E5354 55565758 595A2E2E 2E2E2E2E 30313233 34353637
13794AAC: 38392E2E 2E2E2E2E      0 30313233 34353637 38396162 63646566      0
13794ACC: 202020 20202020 20202828 28282820 20202020 20202020 20202020 20202020
13794AEC: 20881010 10101010 10101010 10101010 10040404 4040404 4040410 10101010
13794B0C: 10104141 41414141 1010101 1010101 1010101 1010101 1010101 10101010
13794B2C: 10104242 42424242 2020202 2020202 2020202 2020202 2020202 10101010
13794B4C: 20000000 436F6D6D 756E6963 6174696F 6E206572 726F7220 6F6E2073 656E6400

```

```

13794B6C: 546F6F20 6D616E79 206C696E 6B730000 426C6F63 6B206465 76696365 20726571
13794B8C: 75697265 64000000 41726720 6C697374 20746F6F 20626967          0 4E6F2073
13794BAC: 75636820 70726F63 65737300 4E6F7420 6F776E65 72000000 4E6F2073 75636820

```

Log buffer:

```

6:21:19 UTC Mon Sep 6 2010
CMD: 'alias exec cas clear auth sess' 06:21:19 UTC Mon Sep 6 2010
CMD: 'alias exec sas show auth sess' 06:21:19 UTC Mon Sep 6 2010
CMD: 'alias exec cpu show proc cpu | inc CPU' 06:21:19 UTC Mon Sep 6 2010
CMD: 'alias exec si show run int gi6/25' 06:21:19 UTC Mon Sep 6 2010
CMD: 'line con 0' 06:21:19 UTC Mon Sep 6 2010
CMD: ' exec-timeout 0 0' 06:21:19 UTC Mon Sep 6 2010
CMD: ' stopbits 1' 06:21:19 UTC Mon Sep 6 2010
CMD: ' speed 38400' 06:21:19 UTC Mon Sep 6 2010
CMD: 'line vty 0 4' 06:21:19 UTC Mon Sep 6 2010
CMD: 'scheduler runtime netinput 100' 06:21:19 UTC Mon Sep 6 2010
CMD: 'mac address-table static 0023.abf8.3303 vlan 1 interface GigabitEthernet6/15'
06:21:19 UTC Mon Sep 6 2010
CMD: 'end' 06:21:19 UTC Mon Sep 6 2010

*Sep 6 06:21:19.103: %SW_VLAN-6-VTP_DOMAIN_NAME_CHG: VTP domain name changed to campus1.
*Sep 6 06:21:21.779: %SYS-5-CONFIG_I: Configured from memory by console
*Sep 6 06:21:21.875: %SYS-5-RESTART: System restarted --
Cisco IOS Software, Catalyst 4500 L3 Switch Software (cat4500e-ENTSERVICES-M),
Experimental Version 12.2(20100723:074204) [././././././ios/sys 179]
Copyright (c) 1986-2010 by Cisco Systems, Inc.
Compiled Mon 06-Sep-10 22:11 by cisco
*Sep 6 06:21:23.363: Slot 0 : delete
*Sep 6 06:21:23.363: K5SuperportSetConfig:
*Sep 6 06:21:23.363: num of Superports : 4, SuperportIds( 57, 57, 57, 57 )
*Sep 6 06:21:23.363: K5SuperportGroupMode Xauik5PortSpeedType 10G10
*Sep 6 06:21:23.363: K5SuperportConfig:
*Sep 6 06:21:23.363: K5SuperportUsageState Populated, 4K5SuperportManagementProtocol
VsiK5HeaderType K10 SCH Preamble, Max SubportId : 9
*Sep 6 06:21:23.363: num of subports : 1, SubportConfig:
*Sep 6 06:21:23.363: SubportConfig( K5SubportId : 0, PimHwPhyportId : 240 )
*Sep 6 06:21:23.711: %C4K_JOB-4-OVERRUN: (Suppressed 1 times)Job Lj-poll ran 20941
microseconds (its runTimeMax was 2000)
*Sep 6 06:21:23.711: Slot 0 : new
*Sep 6 06:21:23.711: K5SuperportSetConfig:
*Sep 6 06:21:23.711: num of Superports : 4, SuperportIds( 57, 57, 57, 57 )
*Sep 6 06:21:23.711: K5SuperportGroupMode Xauik5PortSpeedType 10G10
*Sep 6 06:21:23.711: K5SuperportConfig:
*Sep 6 06:21:23.711: K5SuperportUsageState Populated, 3K5SuperportManagementProtocol
VsiK5HeaderType K10 SCH Preamble, Max SubportId : 9
*Sep 6 06:21:23.711: num of subports : 2, SubportConfig:
*Sep 6 06:21:23.711: SubportConfig( K5SubportId : 8, PimHwPhyportId : 242 )
*Sep 6 06:21:23.711: SubportConfig( K5SubportId : 9, PimHwPhyportId : 243 )CMD: 'en'
06:21:56 UTC Mon Sep 6 2010
CMD: 'plat' 06:21:57 UTC Mon Sep 6 2010
CMD: 'platform cr' 06:21:57 UTC Mon Sep 6 2010
CMD: 'platform crashdump d' 06:21:58 UTC Mon Sep 6 2010
CMD: 'platform crashdump ' 06:21:58 UTC Mon Sep 6 2010

Supervisor (WS-X45-SUP6-E) Board Specific Crash Data:
MCSR: 0x0
L1CSR0: 0x10001 L1CSR1: 0x10001
SRR0: 0x10999e70 CSRR0: 0x0 MCSRR0: 0x0
MCAR: 0x0
ESR: 0x2000000
CISR0: 0x0 CISR1: 0x0
L2CTL: 0xa0000000
L2CAPTDATAHI: 0x0 L2CAPTDATALO: 0x0

```

```
L2CAPTECC: 0x0
L2ERRDET: 0x0
L2ERRDIS: 0x0
L2ERRATTR: 0x0
L2ERRADDRH: 0x0L2ERRADDRL: 0x0
L2_ERRCTL: 0x0
DDR_CAPTURE_DATA_HI: 0x0 DDR_CAPTURE_DATA_LO: 0x0
DDR_CAPTURE_ECC: 0x0
DDR_ERR_DETECT: 0x0
DDR_ERR_DISABLE: 0x0
DDR_ERR_INT_EN: 0x9
DDR_CAPTURE_ATTRIBUTES: 0x0
DDR_CAPTURE_ADDRESS: 0x0
DDR_CAPTURE_EXT_ADDRESS: 0x0
DDR_ERR_SBE: 0xff0000
PCI_ERR_DR: 0x0
PCI_ERR_ATTRIB: 0x0
PCI_ERR_ADDR: 0x0
PCI_ERR_EXT_ADDR: 0x0
PCI_ERR_DH: 0x0PCI_ERR_DL: 0x0
Machine Check Interrupt Count: 0
L1 Instruction Cache Parity Errors: 0
L1 Instruction Cache Parity Errors (CPU30): 0
L1 Data Cache Parity Errors: 0
```

Jawa Crash Data:

```
Interrupt Mask: 0xe180
Interrupt: 0x0
```

```
Galk5DriverMan( 0 )
  SlotType( 3 )
  State( Galk5DriverManStateReady )
  SilentRollRegister( 0 )
  GldMajorVersion( 0 )
  CardRevision( 0 )
  GldMinor( 1)
  Load Dynamic Driver( No )
```

```
Galk5DriverMan( 1 )
  SlotType( 1 )
  State( Galk5DriverManStateReady )
  SilentRollRegister( 0 )
  GldMajorVersion( 0 )
  CardRevision( 0 )
  GldMinor( 1)
  Load Dynamic Driver( No )
```

Switch#

