Managing LightStream 2020 Traps

This chapter tells you how to customize the way your LightStream 2020 multiservice ATM switch (LS2020 switch) displays and logs traps. The procedures in this chapter tell you how to control the fate of individual traps or groups of traps. You can discard traps, store them in a log, or display them on the local console, in a command line interface (CLI) session, or on a network management system (NMS). If you are not familiar with trap filter levels, read the overview in the chapter entitled "About LightStream 2020 Traps."

This chapter provides procedures for

- Managing Individual Traps
- Displaying the Trap Status
- Working with Trap Filter Levels
- Logging Traps

At the end of this chapter is a diagram (Figure 2-5) summarizing trap movement through the LS2020 system and the CLI commands that affect trap flow. Each of the CLI commands in this diagram is explained in detail in this chapter.

Managing Individual Traps

Trap reporting is controlled by trap filter level settings, as discussed in the chapter entitled "About LightStream 2020 Traps." The procedures in this section allow you to override process filter settings for individual traps. The process filter is the first filter a trap encounters, and it decides which traps to pass into the Master Management Agent (MMA). You can instruct the process filter to forward specific traps that it would normally discard. You can also tell it to prevent the MMA from sending specific traps to an NMS or a terminal running a CLI session.

The procedures in this section let you

- Turn a trap on or off globally. Using this procedure, you can turn on specific traps that would otherwise be discarded from the system. For example, if your switch is set to discard all debug and trace traps (the default), you can override that setting and instruct the system to save one or more specific traps.
- Enable or disable a trap in a particular node (globally). Using this procedure, you can turn off specific traps. For example, if your switch is set to send all operational (oper) traps to the NMS, you can disable oper trap numbers that appear too frequently or you deem unimportant. Disabled traps will go to the log and console, but not to the terminal running the CLI session or to the NMS.

Procedures for changing the way the system handles all traps belonging to a particular trap type are in the section entitled "Working with Trap Filter Levels."

Note The console port transmits all traps that are recorded in the trap log. Disabling individual traps does not prevent them from being displayed on the console, but does prevent them from being displayed on the terminal running the CLI session or the NMS.

Procedure for Turning an Individual Trap On or Off

This section tells you how to turn a particular trap on or off. Turning a trap on allows it to be passed to the MMA, even if it has a priority level below the trap filter level set for the process filter.

The default for all traps in all processes is *off*. The process filter determines which traps that are turned off pass to the MMA. For more information, see the section "Working with Trap Filter Levels."

To turn a trap on or off, perform the following steps:

Step 1 Start CLI and enter protected mode:

```
cli> protected
```

Step 2 Enter the protected mode password:

```
Enter password:
```

Step 3 Verify that the target switch is correct:

```
*cli> show chassis general
```

If you need instructions on using the **set SNMP hostname** command to change the target switch, refer to the *LightStream 2020 Network Operations Guide*.

Step 4 Set the SNMP community to a read/write community:

```
*cli> set snmp community <community name>
```

Where

`<community name>` is the SNMP read/write community that you want to access.

Step 5 If you want to turn the trap on or off for a particular process:

```
*cli> walksnmp lwmaTrapCliAlias
```

Find the process you want in the resulting list. Figure 2-1 shows the typical output from this command.

Step 6 Turn a trap on or off using either of the following commands:

If you want the trap on for a single process:

```
*cli> set trap pid{<#|alias>} {on|off} <trap#> [<group name>]
```

If you want the trap on for all processes:

```
*cli> set trap global {on|off} <trap#> [<group name>]
```

Where

- `{<#>|<alias>}` is the process number or alias name.
- `{on|off}` specifies whether the trap is on or off. The default is off.

Figure 2-1 Typical Walksnmp Display

```
cli> walksnmp lwmaTrapCliAlias
Name: lwmaTrapCliAlias.3      Value: CAC
Name: lwmaTrapCliAlias.4      Value: GIDD
Name: lwmaTrapCliAlias.5      Value: NPCC
Name: lwmaTrapCliAlias.6      Value: LCC3
Name: lwmaTrapCliAlias.7      Value: LCC9
Name: lwmaTrapCliAlias.8      Value: LCC5
Name: lwmaTrapCliAlias.10     Value: LCC7
Name: lwmaTrapCliAlias.37     Value: ND
Name: lwmaTrapCliAlias.40     Value: TRAPMON
Name: lwmaTrapCliAlias.45     Value: cardmon
Name: lwmaTrapCliAlias.47     Value: KLOG
Name: lwmaTrapCliAlias.48     Value: NPTMM
Name: lwmaTrapCliAlias.49     Value: COLLECTOR...
                    PID number          Alias name
```

H3463

- `<trap#>` is the trap(s) you want to turn on or off. Enter the symbolic trap name or trap number. You can set multiple trap numbers by specifying the trap numbers (for example, ndd_3 ndd_14 lcc_5), or by using the wild card character **"*"** to specify all traps for a particular process. If you use `*`, you must enclose the expression in which it appears with quotes. For example, to specify all NDD traps, type **show trap "ndd*"**.
- `[<group name>]` is an optional argument that defines a group of traps. For this argument to be used, the group must be defined in an ASCII file called cli.groups in the /usr/app/base/config directory. Refer to the section "Creating the cli.groups File" later in this chapter for instructions on creating the cli.groups file.

Step 7 To display the status of the trap for a single process, enter

```
*cli> show trap pid {<#>|<alias>} "*"
```

To display the status of the trap for all processes:

```
*cli> show trap "*"
```

The status display shows whether a trap is on, off, or disabled. If the status is either on or off, the trap is enabled. Otherwise, the trap is disabled.

At the conclusion of this procedure, each trap that you have turned on passes to the MMA whenever that trap is generated in the selected process. The process filter determines which traps that are turned off pass to the MMA. For more information, see the section "Working with Trap Filter Levels."

Procedure for Enabling/Disabling an Individual Trap

This section tells you how to enable or disable individual traps for all processes. When you disable a trap in an LS2020 switch, the MMA does not send it to an NMS or a terminal running the CLI session, though it will still go into the trap log and be displayed on the console. You may want to disable a trap if it appears frequently and you feel that its display is unnecessary.

To enable or disable a trap, perform the following steps:

Step 1 Start CLI and enter protected mode:

```
cli> protected
```

Step 2 Enter the protected mode password:

```
Enter password:
```

Step 3 Verify that the target switch is correct:

```
*cli> show chassis general
```

If you need instructions on using the **set SNMP hostname** command to change the target switch, refer to the *LightStream 2020 Network Operations Guide*.

Step 4 Set the SNMP community to a read/write community:

```
*cli> set snmp community <community name>
```

Where

`<community name>` is the SNMP read/write community that you want to access.

Step 5 Enable or disable the trap:

```
*cli> set trap {enable|disable} <trap#> [<group name>]
```

Where

- `{enable|disable}` indicates whether the trap is to be enabled or disabled. The default is enabled.
- `<trap#>` identifies the trap(s) that you want to turn on or off. Enter the symbolic trap name or trap number. You can set multiple trap numbers by specifying the trap numbers (ndd_3 ndd_14 lcc_5), or by using the wild card character (`*`) to specify all traps for a particular process. If you use the wild card character (`*`), you must enclose the expression in which it appears with quotes. For example, to specify all NDD traps, you type **`show trap "ndd*"`**
- `[<group name>]` is an optional argument used to define a group of traps that you want to turn on or off. For this argument to be used, the group must be defined in an ASCII file called cli.groups in the /usr/app/base/config directory. Refer to the section "Creating the cli.groups File," later in this chapter, for instructions on creating the cli.groups file.

Step 6 Display the status of each trap in the MMA:

```
*cli> show trap "*"
```

The status display shows whether a trap is on, off, or disabled. If the status is either on or off, the trap is enabled. Otherwise, the trap is disabled. See the section "Procedure for Turning an Individual Trap On or Off" above for a more detailed description of trap handling in a specific process.

After you perform this procedure, the disabled trap for the selected node is neither passed to the CLI nor displayed on a third-party NMS.

Displaying the Trap Status

This section tells you how to view the status of every trap within a particular process or for an MMA. The status display shows traps as either on or off and enabled or disabled. See the section "Procedure for Turning an Individual Trap On or Off," earlier in this chapter, for a description of the on/off state. See the section "Procedure for Enabling/Disabling an Individual Trap," for a description of the enabled/disabled state.

View Status of One or More Traps for a Process

To view the status of one or more traps for a process, perform the following steps:

Step 1 Verify that the target switch is correct:

```
cli> show chassis general
```

If you need instructions on using the **set SNMP hostname** command to change the target switch, refer to the *LightStream 2020 Network Operations Guide*.

Step 2 Set the SNMP community to a read/write community:

```
cli> set snmp community <community name>
```

Where

`<community name>` is the SNMP read/write community that you want to access.

Step 3 View the status of one or more traps for the process:

```
cli> show trap pid {<#>|<alias>} <trap#> [<group name>]
```

Where

- `{<#>|<alias>}` is the number or the alias name of the process. See the section "Procedure for Turning an Individual Trap On or Off" earlier in this chapter for information on obtaining pid numbers and aliases.
- `<trap#>` is the number of the trap(s). Enter the symbolic trap name or trap number. You can set multiple traps by specifying the trap numbers (for example, ndd_3 ndd_14 lcc_5), or by using the wild card character (`*`) to specify all traps for a particular process. If you use the wild card character (`*`), you must enclose the expression in which it appears with quotes. For example, to specify all NDD traps, you type **`show trap "ndd*"`**.
- `[<group name>]` is an optional argument used to define a group of traps for which you want to show status. For this argument to be used, the group must be defined in an ASCII file called cli.groups in the /usr/app/base/config directory. Refer to the section "Creating the cli.groups File," later in this chapter, for instructions on creating the cli.groups file.

View Status of One or More Traps for MMA

To view the status of one or more traps for an MMA, perform the following steps:

Step 1 Verify that the target switch is correct:

```
cli> show chassis general
```

If you need instructions on using the **set SNMP hostname** command to change the target switch, refer to the *LightStream 2020 Network Operations Guide*.

Step 2 Set the SNMP community to a read/write community:

```
cli> set snmp community <community name>
```

Where

`<community name>` is the SNMP read/write community that you want to access.

Step 3 View the status of one or more traps for the MMA:

```
cli> show trap <trap#> [<group name>]
```

Where

- <trap#> is the number of the trap(s). Enter the symbolic trap name or trap number. You can set multiple trap by specifying the trap numbers (for example, ndd_3 ndd_14 lcc_5), or by using the wild card character (*) to specify all traps for a particular process. If you use the wild card character (*), you must enclose the expression in which it appears with quotes. For example, to specify all NDD traps, you type **show trap "ndd*"**.
- [<group name>] is an optional argument that defines a group of traps. For this argument to be used, the group must be defined in an ASCII file called cli.groups in the /usr/app/base/config directory. Refer to the section "Creating the cli.groups File," later in this chapter, for instructions on creating the cli.groups file.

For example, if you now enter the command **show trap ndd_3 ndd_4 ndd_5 ndd_1001**, the status of these traps is displayed (see Figure 2-2).

Figure 2-2 Sample Status Display for Specific Traps

```
*cli>  show trap ndd_3 ndd_4 ndd_5 ndd_1001
Trap NDD_3: off - enabled
Trap NDD_4: off - enabled
Trap NDD_5: off - enabled
Trap NDD_1001: off - enabled
*cli>
```

H3471

If you enter the command **show trap "*",** the status of all traps in the MMA is displayed (see Figure 2-3). Figure 2-3 shows a partial display; several screens of traps are displayed when you issue this command.

Figure 2-3 Sample Status Display for All Traps

```
*cli> show trap *
Trap GENERIC_TEST (1): off - enabled
Trap NDD_1 (3): off - enabled
Trap NDD_2 (4): off - enabled
Trap NDD_3 (5): off - enabled
Trap NDD_4 (6): off - enabled
Trap NDD_5 (7): off - enabled
Trap NDD_6 (8): off - enabled
Trap NDD_7 (9): off - enabled
Trap NDD_8 (10): off - enabled
Trap NDD_1000 (11): off - enabled
Trap NDD_1001 (12): off - enabled
Trap NDD_1002 (13): off - enabled
Trap NDD_2000 (14): off - enabled
Trap NDD_2001 (15): off - enabled
...
```

H3472

Creating the cli.groups File

The cli.groups file defines groups of traps. You can use this file as an argument for the commands described in the sections above entitled "View Status of One or More Traps for a Process" and "View Status of One or More Traps for MMA." If you do not create and maintain this file, you must manually enter each trap number used with those commands.

To create the cli.groups file, perform the following steps:

Step 1 Start CLI and enter protected mode:

```
cli> protected
```

Step 2 Enter the protected mode password:

```
Enter password:
```

Step 3 Escape from the CLI to the LynxOS bash shell:

```
*cli> shell bash
```

Step 4 Move to the /usr/app/base/config directory:

```
LSnode:2# cd /usr/app/base/config
```

Step 5 Invoke the vi editor:

```
LSnode:2# vi cli.groups
```

When the editor opens the file, enter the group names and trap numbers in the format shown below. Each group definition begins with a colon.

```
:<groupname> <trap#> <trap#> ...
:<groupname> <trap#> <trap#> ...
```

Where

- `<groupname>` is a name that defines the group of traps.
- `<trap#>` is the trap numbers within the group.

The contents of your file will be similar to this:

```
:nd_group NDD_1 NDD_2 NDD_3
:lcc_group LCC_3000 LCC_3002
```

Step 6 When you have finished entering the group names and trap numbers in the file, exit the vi editor by pressing the **Esc** key or **^[**, and entering **zz**.

Step 7 Return to the CLI:

```
LSnode:2# exit
```

Step 8 Exit protected mode:

```
*cli> exit
```

Working with Trap Filter Levels

This section explains how to set trap filter levels. Trap filters control which trap types are dropped, saved, or displayed. Because trap filters reside on a trap path, you might need to modify the settings of more than one filter in order to achieve your desired goal. For instance, if your system uses the default setting and you want to view a trace trap for a particular process on the console, you would need to modify the process filter as well as the console filter. For an overview of trap filter levels, refer to the chapter entitled “About LightStream 2020 Traps.”

This section discusses the four trap filters:

- The *process filter* determines which traps are passed from an active software process to the MMA.

- The *console filter* determines which traps are passed from the MMA to the local console for display.
- The *chassis filter* determines which traps are passed from the MMA to the CLI process or to the NMS for display.
- The *CLI session filter* determines which traps are passed to the CLI process for display on the CLI terminal.

By default, the trap filter levels are set so that SNMP and operational traps are displayed, and SNMP, operational, and informational traps are logged. Trace and debug traps are discarded. The default works well for most networks.

Passing Traps Through the Process Filter

Setting the trap filter level for a particular process determines which traps generated by that process will be passed to the MMA. Traps that are passed from the processes into the MMA are, by default, logged in the trap log. (For more information, see the section entitled "Logging Traps.")

The default trap filter level for all processes is *informational*. This level transmits all SNMP, operational, and informational traps to the MMA. This level is appropriate for most applications. If the defaults have not been changed, you would use the following procedure to let in all trace or debug traps for a specific process or to exclude all info or oper traps for a specific process.

Note Increasing the number of traps being reported could result in flooding the node with traps, which could degrade its performance. It is preferable to use the procedures in the section "Managing Individual Traps."

To set the trap filter level for one or more processes, perform the following steps:

Step 1 Verify that the target switch is correct:

```
cli> show chassis general
```

If you need instructions on using the **set SNMP hostname** command to change the target switch, refer to the *LightStream 2020 Network Operations Guide*.

Step 2 Set the SNMP community to a read/write community:

```
cli> set snmp community <community name>
```

Where

`<community name>` is the SNMP read/write community that you want to access.

Step 3 Display a list of the processes:

```
cli> walksnmp lwmaTrapCliAlias
```

This command lists the process identification (PID) numbers and alias names of all the processes running on this LS2020 switch (see Figure 2-1). Select the process you want from the list in the display.

Step 4 Set the trap filter level for a selected process:

```
cli> set pid {<#>|<alias>} traplevel <value>
```

Where:
{<#>|<alias>} is the process number or alias name.

<value> is oper, info (default), trace, or debug

Step 5 Verify the process trap filter level:

```
cli> show pid {<#>|<alias>} traplevel
```

As a consequence of this procedure, the trap filter level for the specified process filter is changed to the specified level. Subsequently, all traps encompassed by this level are passed from the process to the MMA.

Displaying Traps on the Local Console

The trap filter level setting for the console filter in each LS2020 switch determines which traps are displayed on the local console. The default trap filter level for the console filter is *operational*, meaning that only operational and SNMP traps can pass through the chassis. This level is appropriate for most applications.

Note Disabling individual traps by using the **set trap disable** command does not prevent them from being displayed on the console because the console port displays all traps that reach the trap log.

To set the trap filter level for the LS2020 node's local console, perform the following steps:

Step 1 Verify that the target switch is correct:

```
cli> show chassis general
```

If you need instructions on using the **set SNMP hostname** command to change the target switch, refer to the *LightStream 2020 Network Operations Guide*.

Step 2 Set the SNMP community to a read/write community:

```
cli> set snmp community <community name>
```

Where

<community name> is the SNMP read/write community that you want to access.

Step 3 Set the trap filter level for the console or turn off the console display:

```
cli> set chassis consoletraplevel <value>
```

Where

<value> is off (displays no traps), oper, info (default), trace, or debug.

Step 4 Verify that the console trap filter level has been changed:

```
cli> show chassis general
```

When you perform this procedure, the trap filter level for the console is changed to the specified level. Subsequently, the console displays all traps encompassed by this level. Using this procedure, you can also turn off the traps display on the console.

Displaying Traps on the NMS

The trap filter level for the chassis filter determines which traps are passed from the MMA to the NMS and the CLI process. The default trap filter level is *operational*, meaning that, at most, only operational and SNMP traps can pass through the chassis filter.

Traps passed to the NMS from the MMA are displayed on the NMS, unless the NMS has its own filtering capabilities. The traps passed to the CLI process from the MMA must also pass the CLI session filter in order to be displayed on the terminal running the CLI Session (see the following section, "Displaying Traps on the Terminal Running the CLI Session").

Note The following procedure temporarily changes the configured trap filter level setting; the change is lost when the switch is rebooted. To make the change permanent, use the LS2020 Configurator to edit the configuration and update the appropriate node (see the *LightStream 2020 Configuration Guide*).

To temporarily change the configured setting for the chassis trap filter level, perform the following steps:

Step 1 Verify that the target switch is correct:

```
cli> show chassis general
```

If you need instructions on using the **set SNMP hostname** command to change the target switch, refer to the *LightStream 2020 Network Operations Guide*.

Step 2 Set the SNMP community to a read/write community:

```
cli> set snmp community <community name>
```

Where

`<community name>` means the SNMP read/write community that you want to access.

Step 3 Set the trap filter level for the chassis filter:

```
cli> set chassis traplevel <trap value>
```

Where

`<value>` is oper (default), info, trace, or debug.

Step 4 Verify that the trap filter level has been changed:

```
cli> show chassis agent
```

As a consequence of this procedure, the trap filter level for the chassis filter is changed to the specified level. Subsequently, all traps encompassed by this level are sent to the CLI and the NMS.

Displaying Traps on the Terminal Running the CLI Session

Use the following procedure if you wish to display different types of traps on the terminal running the CLI session than on the NMS.

The trap filter level setting for the CLI session filter determines which traps are displayed by the terminal running the CLI session. The default trap filter level for the CLI is *debug*. This means that all traps that passed through the chassis filter (operational and SNMP in the default state) can pass through the CLI session filter. This level is appropriate in most cases.

To set the trap filter level for a CLI session, perform the following steps:

Step 1 Verify that the target switch is correct:

```
cli> show chassis general
```

If you need instructions on using the **set SNMP hostname** command to change the target switch, refer to the *LightStream 2020 Network Operations Guide*.

Step 2 Set the trap filter level for the CLI:

```
cli> set cli traplevel <value>
```

Where

`<value>` is off (displays no traps), oper, info, trace, or debug (default).

Step 3 Verify that the trap filter level for the CLI has been changed:

```
cli> show cli traplevel
```

After you perform this procedure, the trap filter level for the CLI session filter is changed to the specified level. Subsequently, the terminal running the CLI displays all traps encompassed by this level.

Logging Traps

You can log the traps that occur on each LS2020 switch. The traps are stored on the hard disk of the switch in a file called mma.traplog in the /usr/tmp/mma directory. This file can store approximately 6000 traps before the oldest trap is overwritten by the latest trap. The trap filter levels set for each process determine which traps are logged. This section tells you how to enable or disable the trap log.

Enabling or Disabling the Trap Log

You can enable or disable the trap log for a particular LS2020 switch; however, traps cannot be logged unless the trap log is enabled. The default setting for the trap log is enabled (on). This setting is appropriate for most networks.

Note The following procedure temporarily changes the trap log status attribute; any change is lost when the switch is rebooted. To make the change permanent, use the LS2020 Configurator to edit the configuration (see the *LightStream 2020 Configuration Guide)*.

If a node's trap log file is moved or deleted, trap logging is effectively disabled for that node. If the file /usr/tmp/mma/mma.traplog is not present, use this procedure to re-enable trap logging.

To enable or disable the trap log for a particular LS2020 switch, perform the following steps:

Step 1 Verify that the target switch is correct:

```
cli> show chassis general
```

If you need instructions on using the **set SNMP hostname** command to change the target switch, refer to the *LightStream 2020 Network Operations Guide*.

Step 2 Set the SNMP community to a read/write community:

```
cli> set snmp community <community name>
```

Where

<community name> is the SNMP read/write community that you want to access.

Step 3 Enable or disable the trap log for a particular chassis:

```
cli> set chassis traplog <value>
```

Where

<value> is on (enables the trap log) or off (disables the trap log)

Step 4 Verify that the trap log has been enabled or disabled for a particular LS2020 switch:

```
cli> show chassis agent
```

After you perform this procedure, the trap log for the specified chassis is enabled or disabled.

Viewing the Trap Log

Any traps passed from the software processes to the MMA are recorded in a circular file named /usr/tmp/mma/mma.traplog. This section tells you how to view the trap log from the LynxOS shell and view the trap log from the CLI.

Viewing the Trap Log from a LynxOS Shell

If you are working in the LynxOS shell, you can view the trap log file by entering a command of the following form:

```
cbufpr [-hv] [-all] [-tail] <-number> [-f] [-stat] <-level> /usr/tmp/mma/mma.traplog | more
```

Where

- [h] displays a help message. Other arguments with the **-h** argument are ignored.
- [v] displays cbufpr (trap log) version information. Other arguments with the **-v** argument are ignored (except the **-h**).
- [all] allows you to read files of all formats, including files that are not circular.
- [tail] is an optional argument that displays the last 20 lines of the file (the lines containing the most recent traps). If you do not enter this argument, the entire file is displayed.
- <number> specifies the number of lines to display. This switch can be used with the **-tail** switch to specify the number of lines displayed from the bottom of the file.
- [f] continues reading from the end of file rather than exiting. The switch allows you to display traps as they accumulate while you are viewing other parts of the file. Enter **^C** (Control-C) to kill the process.
- [stat] reports the current position of the write pointer.
- <level {snmp | oper | info | trace | debug}> reports traps at and above the indicated level.
- | more displays one page at a time. Press the space bar to display the next page. If you do not use | more, the file scrolls across the screen.

For more information on the **cbufpr** command, see the *LightStream 2020 NP O/S Reference Manual*.

Viewing the Trap Log from the CLI

If you are working in the CLI, you can view the trap log by entering the following at the CLI prompt:

```
cli> show file traplog
```

You can use the optional **-tail** argument with the **show file** command to display the last 20 (or so) lines of the trap log file. For more information on the **show file** command, refer to the *LightStream 2020 CLI Reference Manual*.

Moving the Trap Log from the NP

If you are working in the CLI, you can use the following procedure to move the trap log to another NP for viewing on that system. If you are working in the LynxOS shell, use the ftp command.

Note Before attempting to move the trap log from one NP to another, obtain a user name and password for an account on the workstation or host where you intend to place the trap log file.

To move the trap log file to another NP for viewing, perform the following steps:

Step 1 Enter CLI protected mode (required for execution of the **shell** command):

```
cli> protected
```

Step 2 Enter the protected mode password:

```
Enter password:
```

Step 3 Unwind the circular log file:

```
*cli> shell "cbufpr /usr/tmp/mma/mma.traplog > tmp/traplog"
```

The trap log file is a temporary file.

Step 4 Shell out of CLI:

```
*cli> shell "ftp <IP address of destination workstation or host>"
```

You are prompted to log in to the workstation or host.

Step 5 Log in to the workstation or host.

Step 6 You can place the trap log file in any directory other than the login directory on the workstation or host. Enter **cd** <directory name> to change to the appropriate working directory.

Step 7 Place the log file in the directory:

```
ftp> put /tmp/traplog [<new name>]
```

Where

`[<new name>]` is the file name identifying the chassis or the appropriate directory name for the file. For example, if you are moving a trap log for a switch called Light5, the new name could be mma_Light5.traplog.

This command sends the log file to the specified workstation or host. The system tells you when the file transfer is complete.

Step 8 Exit FTP:

```
ftp> quit
```

Use any **more** or **cat** command or a screen editor such as emacs or vi to view the mma.traplog file on the workstation or host.

Step 9 Remove the temporary trap log file:

```
*cli> shell "rm /tmp/traplog"
```

Figure 2-4 shows an example of a trap log. Traps without a switch name have been generated by the local node. Traps that include a switch name have been generated by another LS2020 node and reported to the local node.

Figure 2-4 Trap Log Example

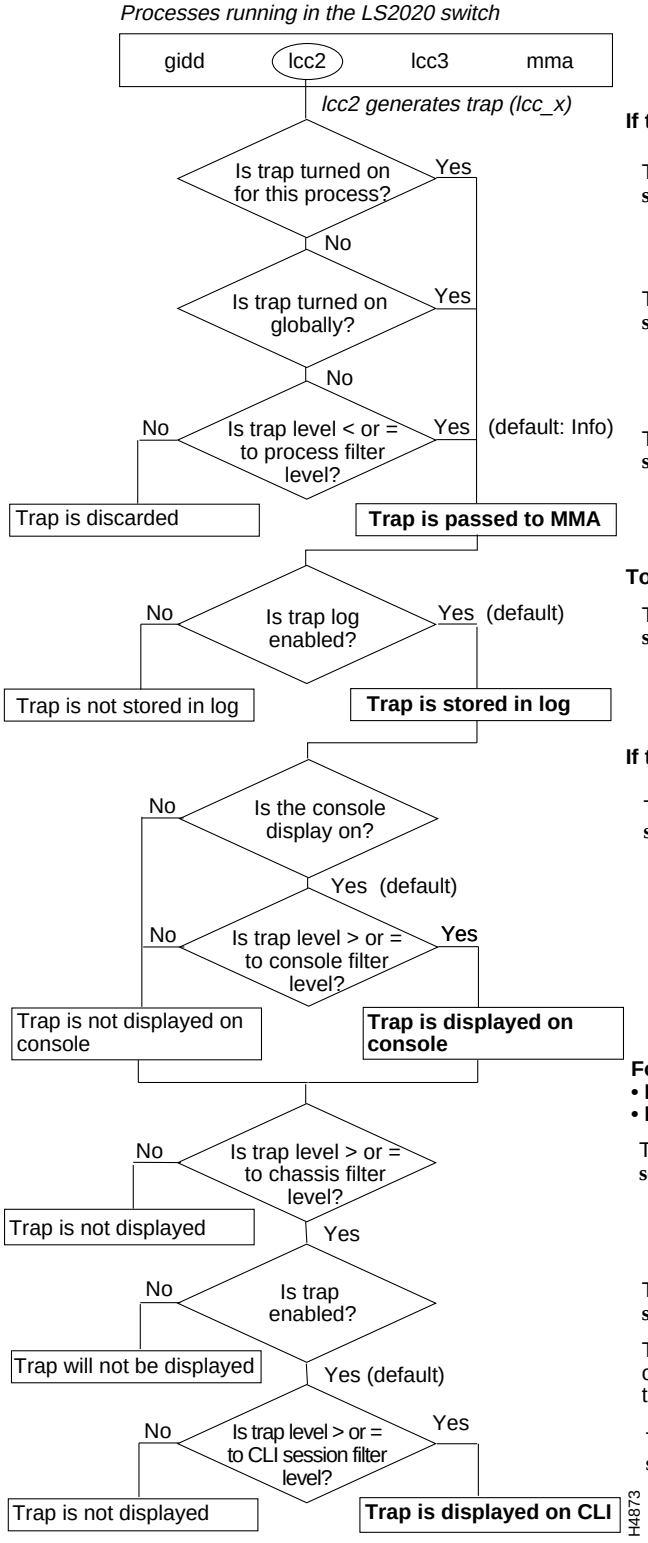
```
(OPER) NPTMM_6 at 08/26/94 14:02:51 EDT (08/26/94 18:02:51 GMT)
TEMPERATURE#2 (103.515F) of card 1 is outside of the normal range
Link Down Trap at 08/26/94 14:03:40 EDT (08/26/94 18:03:40 GMT)
Port 5001
Trap from Light1, System Up Time:  3 Hr 27 Min 23 Sec
(OPER) LCC 12 at 08/26/945 15:01:30 EDT (02/07/95 19:01:30 GMT)
Node Light1 port 5007 entering internal loop mode
...
```

H3468

Summary of Trap Commands

Figure 2-5 provides an overview of the trap movement and commands in the LS2020 system.

Figure 2-5 Overview of Trap Movement and CLI commands



If the trap doesn't reach the MMA, it is discarded.

To turn trap off/on
set trap id {<#> | <alias>} on | off <trap #>} <group name>

To turn trap off/on globally
set trap global on | off <trap #> <group name>

To change process filter level
set pid <pid #> traplevel <value>

To save a trap, you need a trap log.

To enable trap log (/usr/tmp/mma/mma.traplog)
set chassis trap log on

If the trap is logged, it is usually displayed on the console.

To turn console display off
set chassis consoletraplevel off

To change console filter level
set chassis consoletrap level <value>

For a trap to be displayed on the CLI or NMS, it must
• Be greater than the filter level to which the chassis is set
• Be enabled

To change chassis filter level
set chassis traplevel <value>

To enable/disable trap
set trap {enable | disable} <trap #><group name>

The trap will be displayed on a third-party NMS if one has been configured. For more information, see the *LightStream 2020 Installation Guide*.

To change chassis filter level
set cli traplevel <value>