CHAPTER 2

Managing LightStream 2020 Traps

This chapter allows you to customize the way your LightStream 2020 enterprise ATM switch displays and logs traps. The procedures in this chapter provide you with a great deal of flexibility in trap management. Use these procedures to:

- Specify a threshold level of traps to be reported (operational, informational, trace, or debug)
- Specify whether the traps will be displayed
- Select where the traps will be displayed
- View trap displays and log files
- Enable or disable a trap log
- Move the trap log from the NP
- Manage individual traps

Figure 2-1, Figure 2-2, and Figure 2-3 show how traps flow through the LightStream 2020 node. They also show the CLI commands that you can use to affect the flow of traps. Each of these commands is explained in more detail within this chapter.

Figure 2-1 How traps are passed through the system

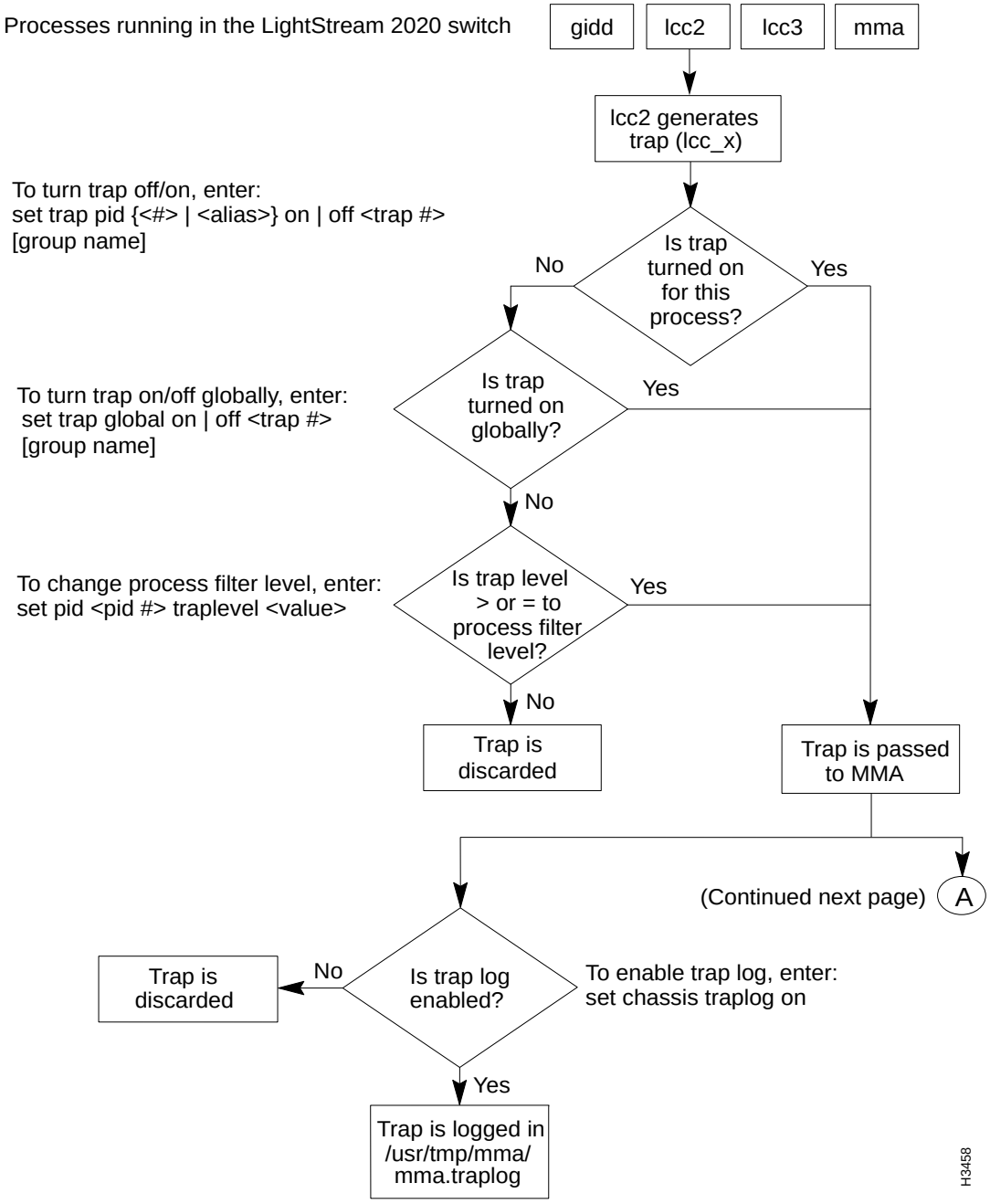


Figure 2-2 How traps are passed through the system (Continued)

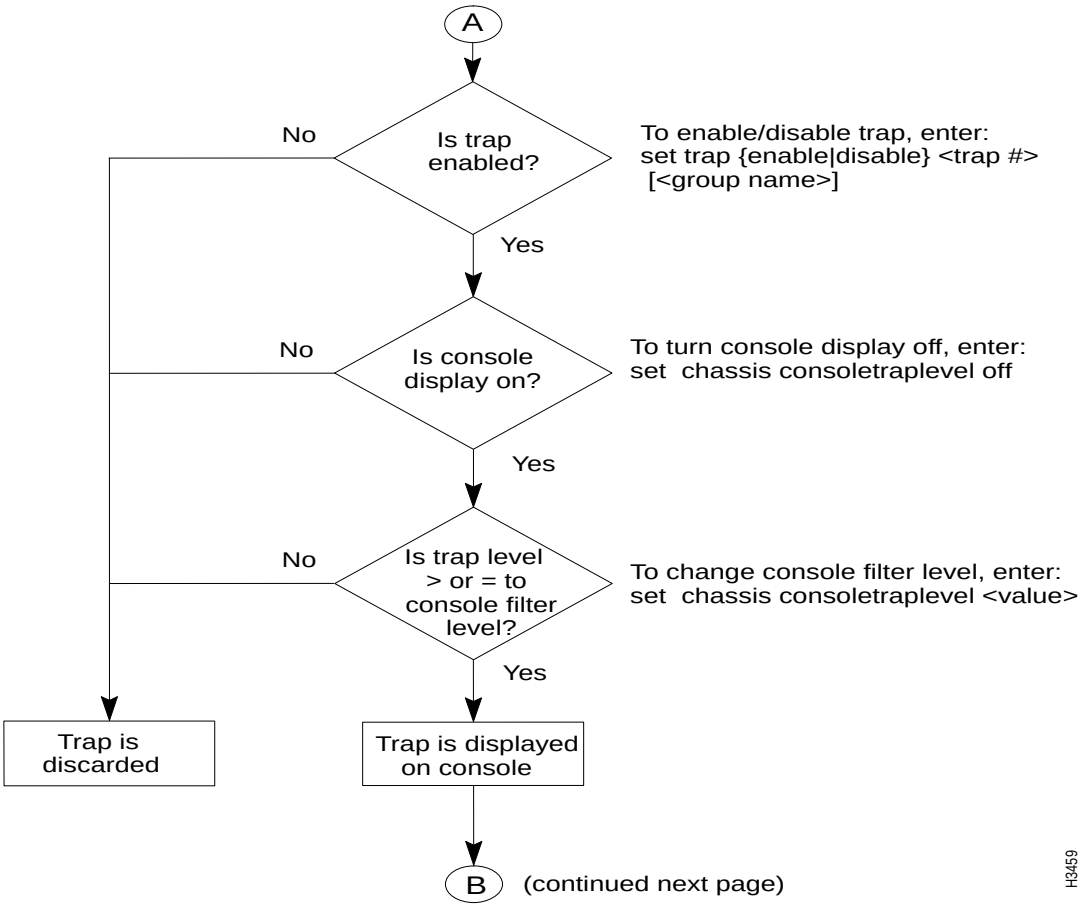
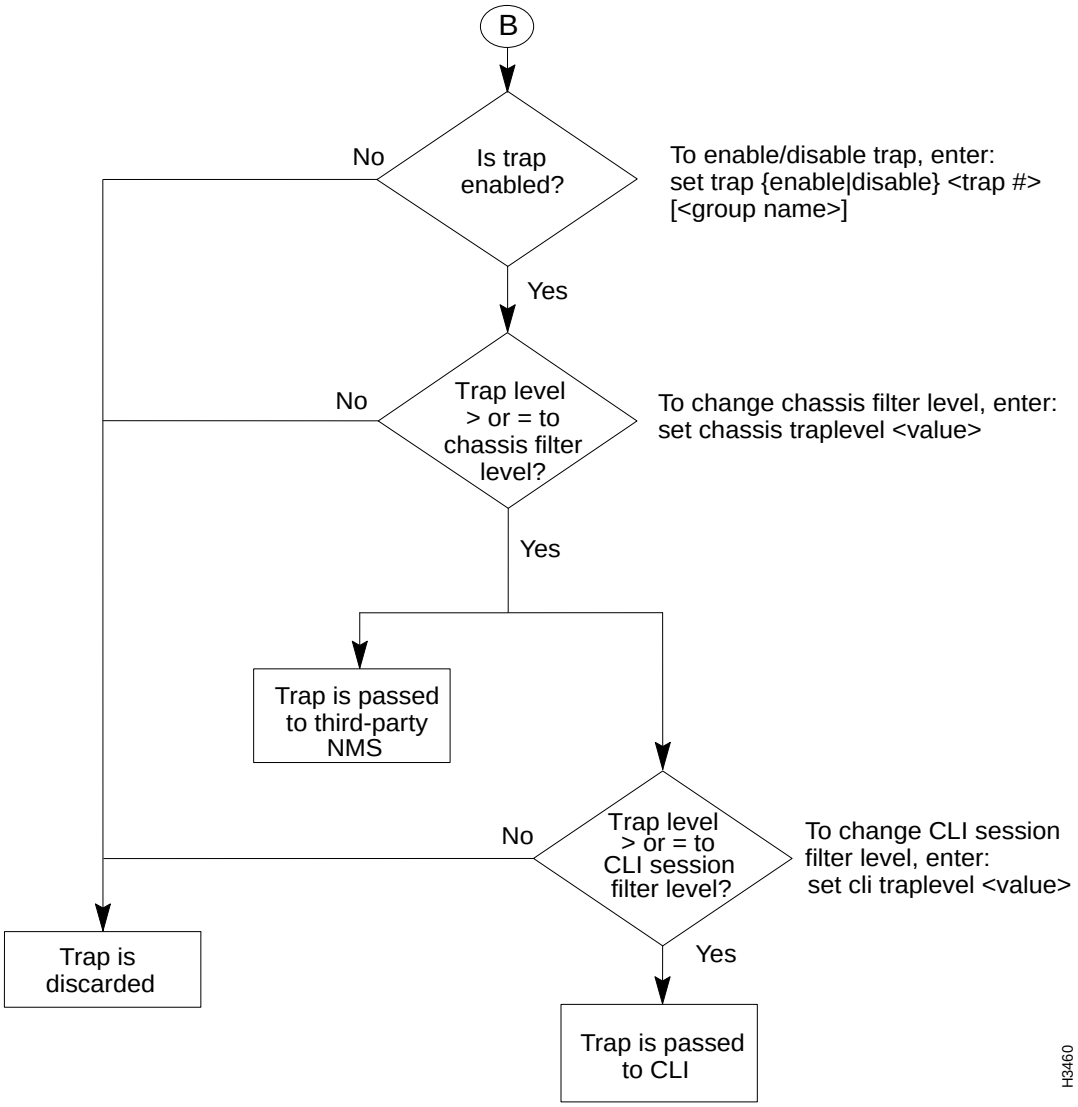


Figure 2-3 How traps are passed through the system (Concluded)



Setting the Trap Reporting Threshold

You determine which traps are displayed by setting the trap reporting threshold, or trap level. There are four trap reporting threshold settings: operational, informational, trace, and debug.

Table 2-1 shows the effect of setting the trap reporting threshold at different levels. As indicated in the table, if you enable a certain trap level, then traps at and above that priority are reported. For instance, if you set the trap reporting threshold to trace, trace traps and all higher priority traps (informational, operational, and SNMP) are reported; debug traps are not reported.

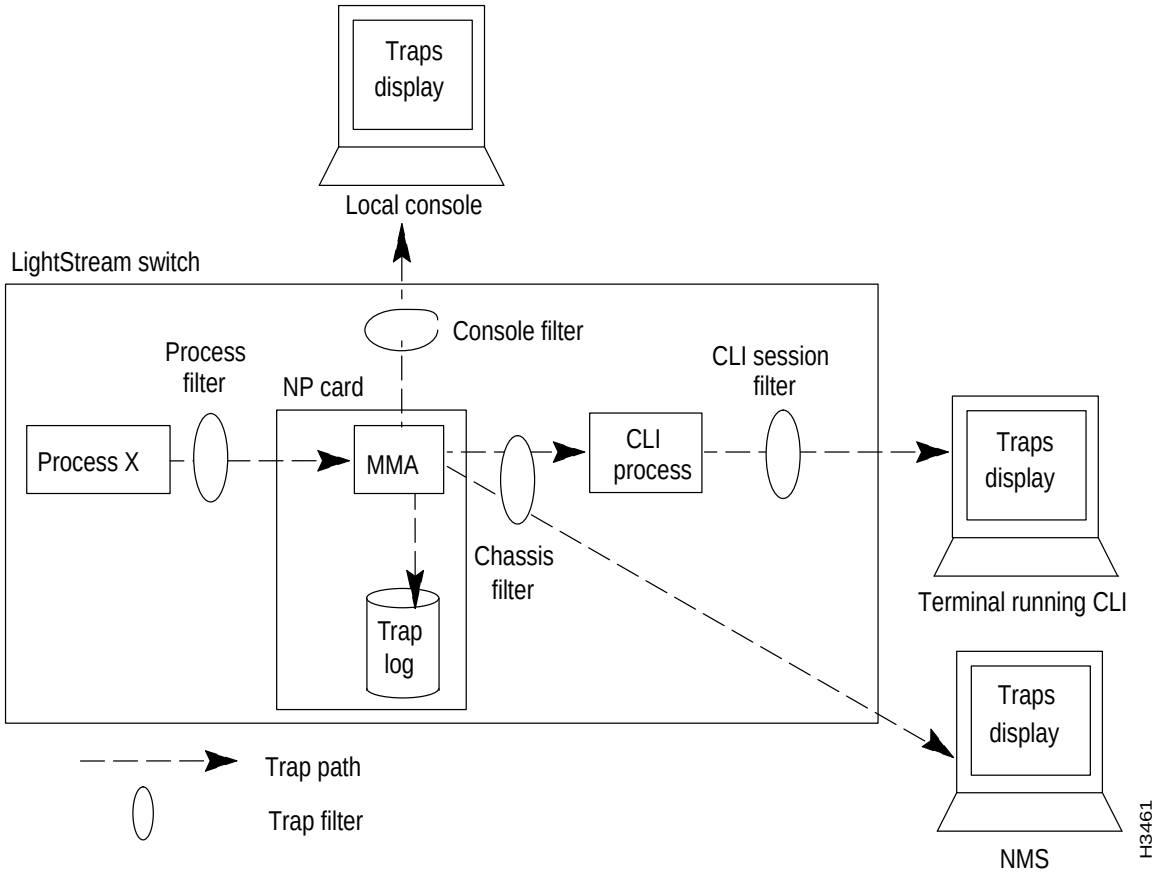
Table 2-1 Trap Reporting Thresholds

	Traps Reported				
Trap Reporting Threshold	**OPER**	**INFO**	**TRACE**	**DEBUG**	**SNMP**
Operational	x				x
Informational	x	x			x
Trace	x	x	x		x
Debug	x	x	x	x	x

Note SNMP traps are always reported. This is not configurable.

Figure 2-4 identifies the different places in the system where you can set the trap reporting threshold. In essence, setting the trap reporting threshold creates a trap filter that either passes the trap to the next process in the system or drops it. At each trap filter, you can set the reporting threshold to operational, informational, trace, or debug.

Figure 2-4 By setting the trap reporting threshold at a particular trap filter, you can prevent traps from being passed to the next process



As shown in Figure 2-4, you can filter traps:

- For each software process running on the switch — The process trap filter determines which traps are passed from the process to the MMA.
- For the chassis —The chassis trap filter determines which traps are passed from the MMA to the CLI process or the NMS.
- For the CLI session — The CLI session trap filter determines which traps are passed from the CLI process for display on the CLI.
- For the console — The console trap filter determines which traps are passed from the MMA to the console.

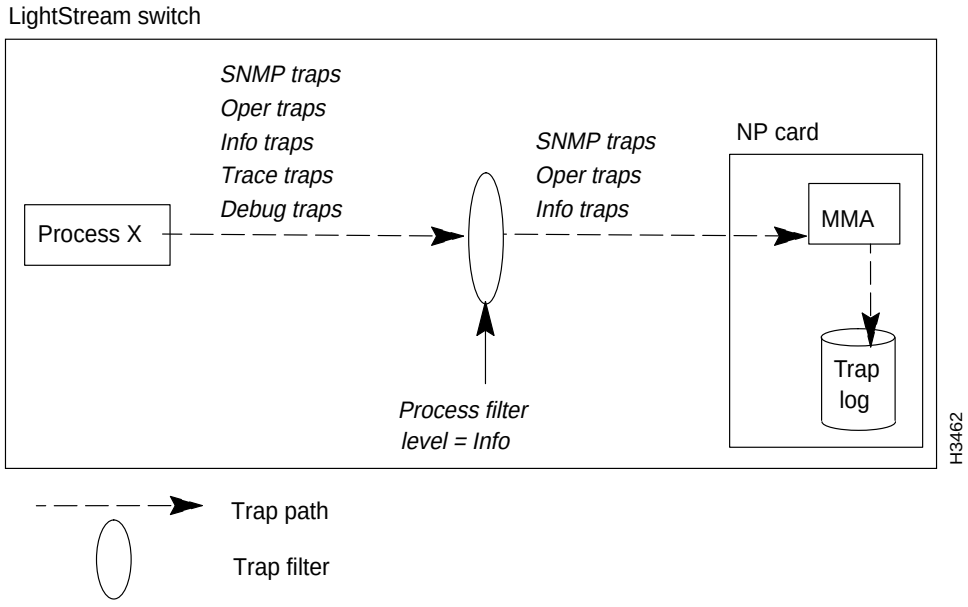
The following sections describe how you can change the various trap filter settings. However, the default settings are appropriate for most networks.

Setting the Trap Reporting Threshold for Processes

Setting the trap reporting threshold for a particular process determines which traps generated by that process are passed to the MMA. Traps that are passed from the processes into the MMA are, by default, logged in the trap log. The default trap reporting threshold for all processes is informational. This level is appropriate for most applications. (Note that the console port transmits all traps that are recorded in the trap log.)

Figure 2-5 shows what happens when a process filter is set to the default, informational.

Figure 2-5 Traps of all priority levels are generated by the software process. However, only SNMP, operational, and informational traps are passed to the MMA.



Follow the procedure below to set the trap reporting threshold for processes.

Procedure

Step 1 Verify that the target switch is correct by entering the following at the `cli>` prompt:

```
cli> show chassis general
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Network Operations Guide*.

Step 2 Set the SNMP community to a read/write community by entering the following at the `cli>` prompt:

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

where

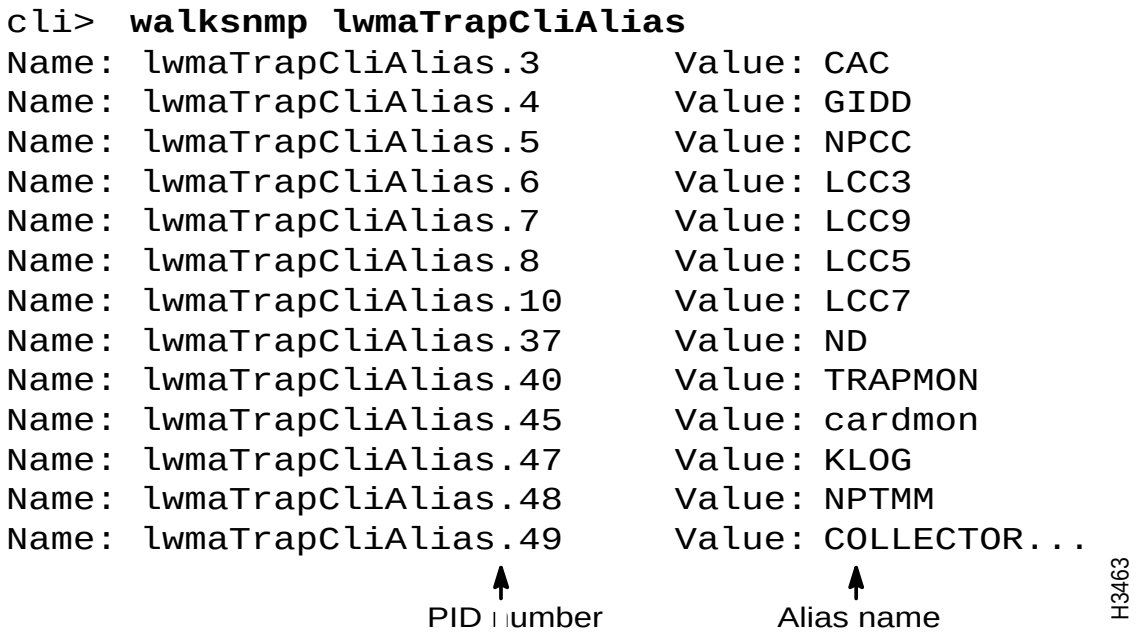
`<community name>` = The SNMP read/write community that you want to access.

Step 3 Enter the following at the `cli>` prompt to display a list of the processes:

```
cli> walksnmp lwmaTrapCliAlias
```

This command lists the PID (process identification) numbers and alias names of all the processes running on this LightStream 2020 switch. The PID numbers follow the term `Name: lwmaTrapCliAlias` and the alias names follow the term `Value`, as shown in Figure 2-6.

Figure 2-6 Typical walksnmp display.



Select the process(es) you want from the list.

Step 4 To set the trap reporting threshold for a selected process, enter the following at the `cli>` prompt:

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

where

`{<#>|<alias>}` = The process number or alias name.

`<value>` = oper
info (default)
trace
debug

Step 5 Verify the process trap reporting threshold by entering the following at the `cli>` prompt:

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

Expected Results

The trap reporting threshold for the specified process filter is changed to the level you specified. All traps at that new level (and higher) are passed from the process to the MMA.

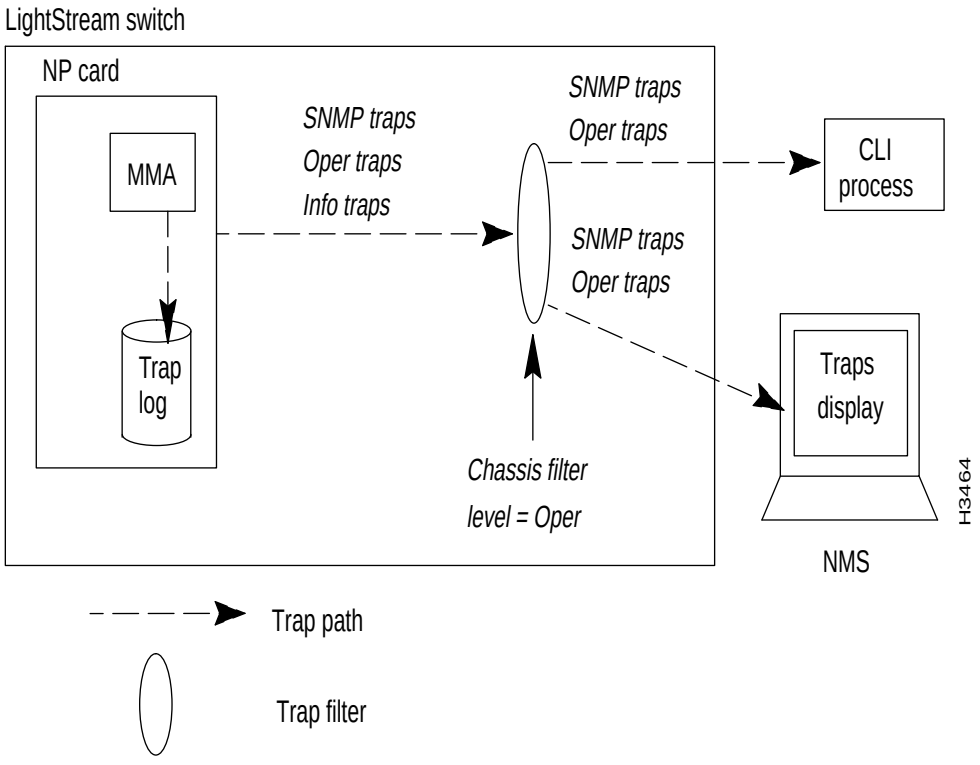
Setting the Trap Reporting Threshold for the Chassis

The trap reporting threshold for the chassis determines which traps are passed from the MMA to the CLI process and to the NMS. The default trap reporting threshold for the chassis is operational. This level is appropriate for most networks.

Figure 2-7 shows what happens when the chassis filter is set to the default, operational, and the process filter has been set to its default, informational.

Note The traps that are passed to the CLI process from the MMA must also pass the CLI session filter (see the section that follows, "Setting the Trap Reporting Threshold for a CLI Session") to be displayed. The traps passed to the NMS from the MMA are displayed on the NMS, unless the NMS has its own filtering capabilities.

Figure 2-7 Although SNMP, operational, and informational traps are received by the MMA, only SNMP and operational traps are passed to the CLI process and the NMS because the chassis filter is set to operational.



Note The trap reporting threshold for the chassis is normally set during network configuration. If you want to temporarily change the configured setting, use the procedure below. (The change will be lost if the switch reboots.) If you want to make a permanent change to the Trap Filter attribute, use the LightStream 2020 configurator to edit the configuration and update the appropriate node, as described in the *LightStream 2020 Configuration Guide*.

Step 1 Verify that the target switch is correct by entering the following at the `cli>` prompt:

```
cli> show chassis general
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Network Operations Guide*.

Step 2 Set the SNMP community to a read/write community by entering the following at the `cli>` prompt:

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

where

`<community name>` = The SNMP read/write community that you want to access.

Step 3 Set the trap reporting threshold for the chassis filter by entering the following at the `cli>` prompt:

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

where

`<value>` = oper (default)
info
trace
debug

Step 4 Verify that the trap reporting threshold has been changed by entering the following at the `cli>` prompt:

```
cli> show chassis agent
```

Expected Results

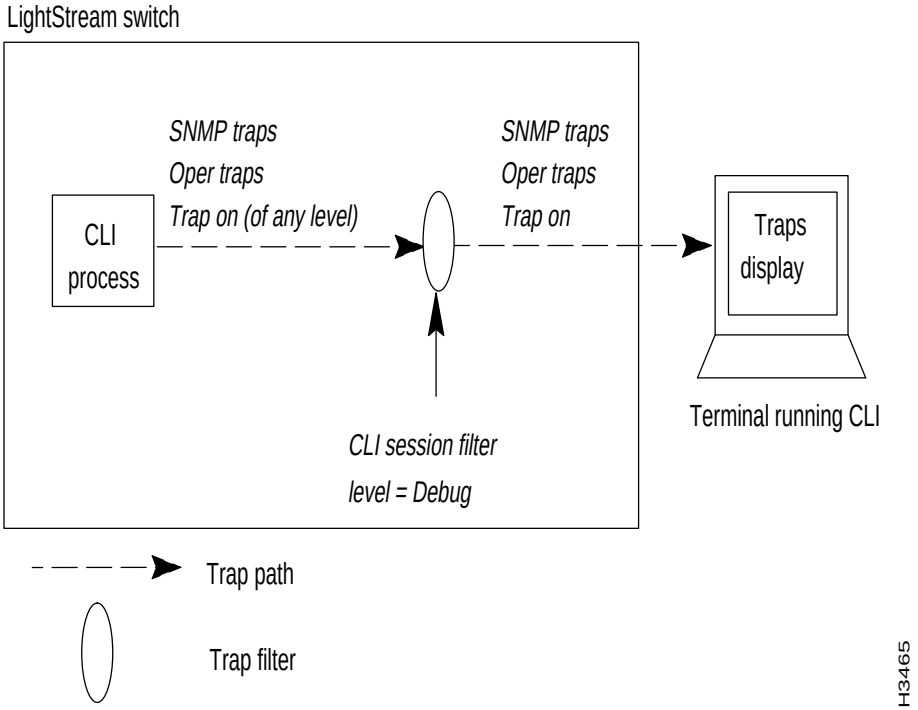
The trap reporting threshold for the chassis filter is changed to the level you specified. All traps at that new level (and higher) are sent to the CLI and the NMS.

Setting the Trap Reporting Threshold for a CLI Session

The trap reporting threshold setting for the CLI session determines which traps are displayed by the terminal running the CLI session. The default trap reporting threshold for the CLI is debug. This level is appropriate for most applications.

Figure 2-8 shows what happens when the CLI session filter is set to the default, debug, and the process and chassis filters have also been set to their defaults.

Figure 2-8 **All traps reported to the MMA are displayed in the CLI. Setting the default to debug ensures that any trap (regardless of its level) is displayed in the CLI. This is important because you can override the filter settings on an individual trap basis (by turning it on) using the procedures in the section "Managing Individual Traps".**



Note You can also turn off the traps display on the terminal running the CLI using this procedure.

Procedure

Step 1 Verify that the target switch is correct by entering the following at the `cli>` prompt:

```
cli> show chassis general
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Network Operations Guide*.

Step 2 Enter the following at the `cli>` prompt to set the trap reporting threshold for the CLI:

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

where

`<value>` = off — displays no traps
oper
info
trace
debug (default)

Step 3 Verify that the trap reporting threshold for the CLI has been changed by entering the following at the `cli>` prompt:

```
cli> show cli traplevel
```

Expected Results

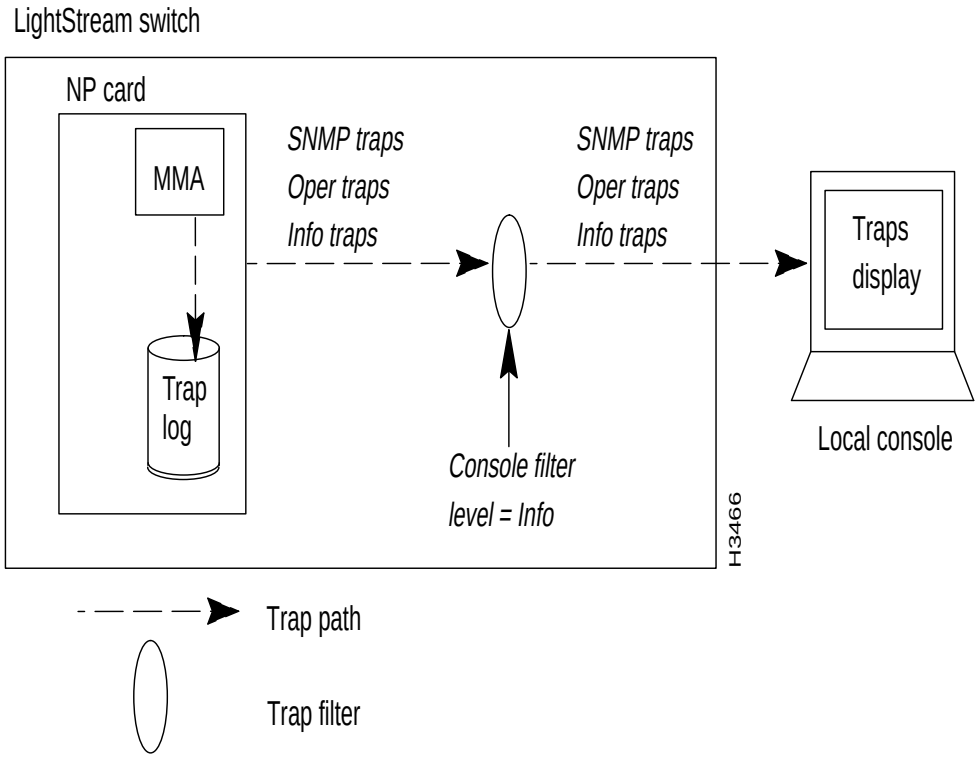
The trap reporting threshold for the CLI session filter is changed to the level you specified. All traps at that new level (and higher) are displayed by the CLI.

Setting the Trap Reporting Threshold for the Console

The trap reporting threshold setting for the console filter in each LightStream 2020 switch determines which traps are displayed on the local console (if the node has a local console). By default, traps are displayed automatically whenever a console is started. The default trap reporting threshold for the console filter is informational. This level is appropriate for most applications. (Note that the console port transmits all traps that are recorded in the trap log; disabling individual traps does not prevent them from displaying on the console.)

Figure 2-9 shows what happens when the console filter is set to the default, informational, and the process filter has also been set to its default, informational.

Figure 2-9 All SNMP, operational, and informational traps passed to the MMA are displayed on the local console.



Follow the procedure below to set the trap reporting threshold for the filtering mechanism.

Note You can also turn off the traps display on the console using this procedure.

Procedure

Step 1 Verify that the target switch is correct by entering the following at the `cli>` prompt:

```
cli> show chassis general
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Network Operations Guide*.

Step 2 Set the SNMP community to a read/write community by entering the following at the `cli>` prompt:

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

where

`<community name>` = The SNMP read/write community that you want to access.

Step 3 Set the trap reporting threshold for the console or turn off the console display by entering the following at the `cli>` prompt:

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

where

`<value>` = off — displays no traps
oper
info (default)
trace
debug

Step 4 Verify that the console trap reporting threshold has been changed by entering the following at the `cli>` prompt:

```
cli> show chassis general
```

Expected Results

The trap reporting threshold for the console is changed to the level you specified. All traps at that new level (and higher) are displayed by the console.

Viewing Traps

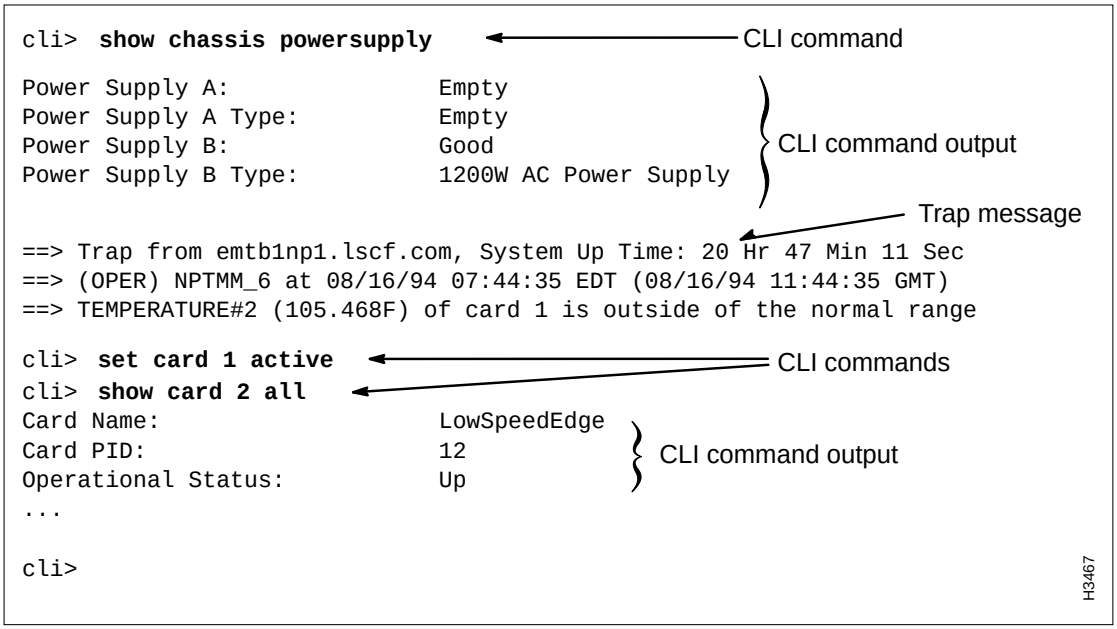
There are three places that it is possible to view traps:

- On a local console attached to the switch
- On a terminal running the CLI
- On an NMS

On the LightStream 2020 console or in the CLI, traps are displayed in the format described in the section “Trap Types” in the chapter “About LightStream 2020 Traps.” Trap messages may be interleaved with the other information displayed by the CLI, depending on how your system is set up. Figure 2-10 shows a typical CLI display including traps.

If you use an NMS, the message format may vary from the format described earlier, but the content is the same. (Refer to the NMS documentation for information on viewing traps there.)

Figure 2-10 Typical CLI traps display.



Logging Traps

You can log the traps that occur on each LightStream 2020 switch. The traps are stored on the NP of the switch in a file called mma.traplog in the /usr/tmp/mma directory. This circular file can store approximately 6000 traps before the oldest trap is overwritten by the newest trap. Which traps are logged is determined by the trap reporting thresholds set for each process. This section tells you how to enable, disable, and view the trap log.

You may also be able to log traps at the NMS. Refer to the NMS documentation for more information.

Enabling or Disabling the Trap Log

This section tells you how to enable or disable the trap log for a particular LightStream 2020 switch. 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.

You usually specify whether the trap log is enabled or disabled during network configuration. If you want to temporarily change the configured setting, use the procedure below. (The change is lost if the system is rebooted.) If you want to make a permanent change to the Trap Log Status attribute, use the LightStream 2020 configurator to edit the configuration and update the appropriate node, as described in the *LightStream 2020 Configuration Guide*.

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

Caution If you disable the trap log for a particular switch, you will not have a record of the traps that were reported if a problem occurs.

Procedure

Step 1 Verify that the target switch is correct by entering the following at the `cli>` prompt:

```
cli> show chassis general
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Network Operations Guide*.

Step 2 Set the SNMP community to a read/write community by entering the following at the `cli>` prompt:

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

where

`<community name>` = The SNMP read/write community that you want to access.

Step 3 Enable or disable the trap log for a particular chassis by entering the following at the `cli>` prompt:

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

where

`<value>` = on—enables the trap log (default)
off—disables the trap log

Step 4 Verify that the traplog has been enabled or disabled for a particular chassis by entering the following at the `cli>` prompt:

```
cli> show chassis agent
```

Expected Results

The trap log for the specific chassis is enabled or disabled, as specified.

Viewing the Trap Log

Any traps passed from the software processes to the MMA are recorded in a circular file, /usr/tmp/mma/mma.traplog. This section tells you how to view the trap log from LynxOS, view the trap log from the CLI, and copy the file to another workstation so you can view it there.

Procedure 1: Viewing the Trap Log from LynxOS

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

```
LSnode:2$ 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 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]` = 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 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. Type **^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 will scroll across the screen.

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

Procedure 2: 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 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 system. If you are working in the LynxOS shell, you can just use **ftp**.

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

Step 1 To enter protected mode, needed to execute the **shell** command, at the `cli>` prompt, enter:

```
cli> protected
```

Step 2 Enter the protected mode password when you see the following prompt:

```
Enter password:
```

Step 3 To unwind the circular log file, type the following at the `*cli>` prompt:

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

The traplog file is a temporary file.

Step 4 At the `*cli>` prompt, enter:

```
*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 To 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 correct working directory.

Step 7 Enter the following at the `ftp>` prompt:

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

where

`[<new name>]` = 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 Enter the following at the `ftp>` prompt:

```
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 To remove the temporary traplog file, type

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

Expected Results

Figure 2-11 shows an example of a trap log. Traps with no switch name have been generated by the local node. Traps that include a switch name have been generated by another LightStream 2020 node and reported to the local node.

Figure 2-11 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

Managing Individual Traps

You can control the reporting of individual traps. This gives you greater flexibility in choosing which traps are reported. Using the capabilities provided in this area, you can:

- Turn a trap on or off in an instance of a particular process. Turning a trap on allows the individual trap to override the trap reporting threshold of the process filter so the trap is reported to the MMA even if its level is below that of the process filter.
- Turn a trap on or off in all instances of a process (turn a trap on or off globally). Turning a trap on globally has the same effect as turning on an individual trap, except that it works for that trap in the case that multiple instances of the same process are running in the LightStream 2020 node.
- Enable or disable a trap in a particular node (globally). Disabling a trap causes it to be dropped by the MMA.

These capabilities are especially useful for advanced debugging and troubleshooting. Because it is anticipated that only experienced users need this functionality, the commands associated with individual traps must be executed from the CLI in protected mode.

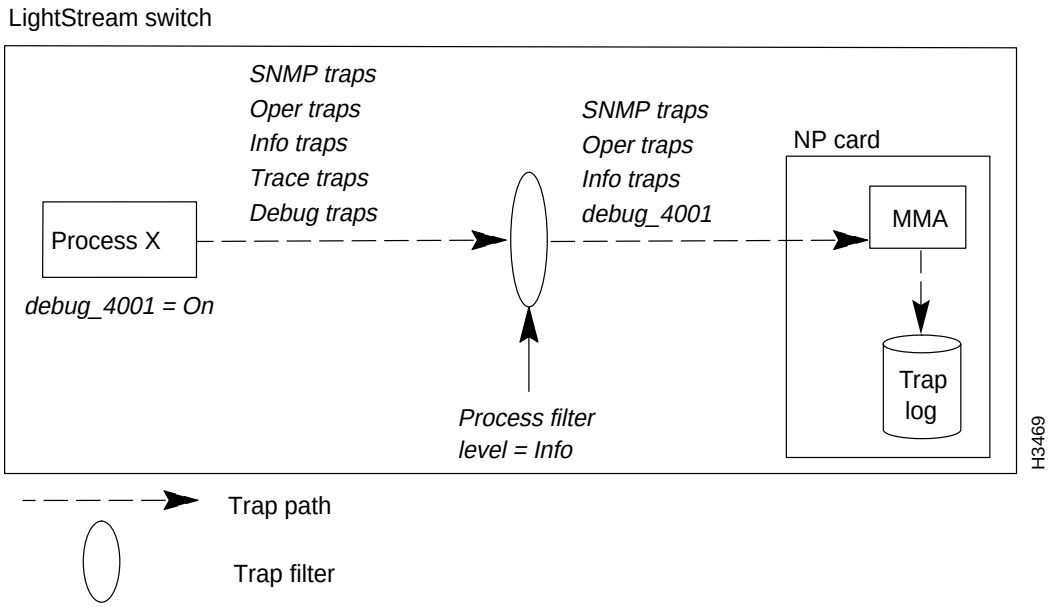
Note The console port transmits all traps that are recorded in the trap log; disabling individual traps does not prevent them from displaying on the console.

Turning a Trap On or Off in a Specific Process

This section tells you how to turn a particular trap on or off in a specific process. Turning a trap on in a particular process allows it to be passed to the MMA, even if it has a severity level below the trap reporting threshold set for the process. This allows you to report a *particular* trap without reporting all traps at that level. This procedure is especially important during troubleshooting and debugging.

Figure 2-12 shows the effect of turning a trap on when the process filter is set to the default, informational.

Figure 2-12 Turning a trap on overrides the trap reporting threshold of the process filter. If the filter is set to informational, and you turn on a particular debug trap, then that debug trap is also passed to the MMA.



In some cases you may have multiple instances of the same process running. Therefore, you can also turn a trap on globally so that it affects the trap in each instance of the process.

The default for all traps in all processes is off. When a trap is set to *off* (the default), the trap is passed to the MMA only if its severity level is equal to or higher than the trap reporting threshold setting of the process filter.

Procedure

Step 1 At the `cli>` prompt, enter:

```
cli> protected
```

Step 2 Enter the protected mode password when you see the following prompt:

```
Enter password:
```

Step 3 Verify that the target switch is correct by entering the following at the `cli>` prompt:

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

If you need instructions on changing the target switch, refer to the *LightStream 2020 Network Operations Guide*.

Step 4 Set the SNMP community to a read/write community by entering the following at the `*cli>` prompt:

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

where

`<community name>` = The SNMP read/write community that you want to access.

Step 5 To determine which processes are running on this node, enter the following at the `*cli>` prompt:

```
*cli> walksnmp lwmaTrapCliAlias
```

Find the process you want in the resulting list. Figure 2-6 shows the output from this command.

Step 6 Turn a trap on or off as follows:

For a trap in one instance of a process, at the `*cli>` prompt, enter:

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

Or for a trap in multiple instances of a process (globally), at the `*cli>` prompt, enter:

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

where

- `{<#>|<alias>}` = The process number or alias name.
- `{on|off}` Specifies whether the trap is on or off. The default is off.
- `<trap#>` = 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 (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 surround the expression in which it appears in quotes. For example, to specify all NDD traps, you would type **show trap "ndd*"**.
- `[<group name>]` = An optional argument that defines a group of traps. To use this argument, 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 each trap in the selected process, enter the following at the `*cli>` prompt:

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

Expected Results

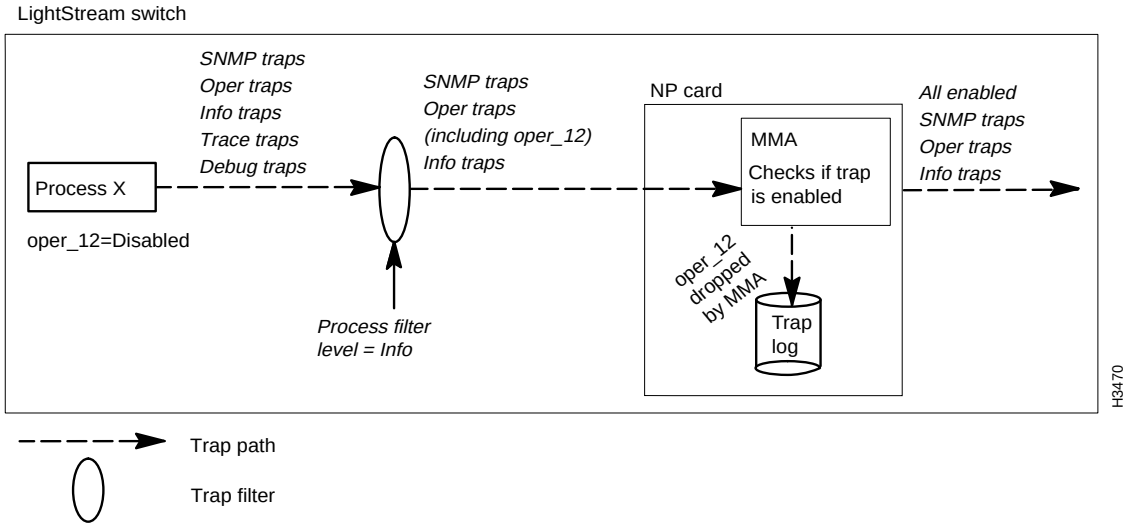
The trap(s) that you have turned on pass to the MMA whenever that trap is generated in the selected process. Traps that are turned off pass to the MMA only if they have an equal or higher severity level than the process filter.

Enabling/Disabling a Trap for All Processes

This section tells you how to enable or disable individual traps for all processes. (Compare this with the previous procedure, which lets you enable or disable a trap for one particular process.) When you disable a trap in a switch, the MMA discards it rather than passing it on to the display device. You may want to disable a trap if it recurs regularly and you feel that its display is unnecessary.

Figure 2-13 shows what happens when traps are enabled, which is the default.

Figure 2-13 The MMA checks to see if the trap is enabled or disabled. If the trap is enabled and its severity level is equal to or greater than the CLI session and console filters, the trap is passed to the CLI and the NMS. If the trap is disabled, it is dropped.



Procedure

Step 1 At the `cli>` prompt, enter:

```
cli> protected
```

Step 2 Enter the protected mode password when you see the following prompt:

```
Enter password:
```

Step 3 Verify that the target switch is correct by entering the following at the `*cli>` prompt:

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

If you need instructions on changing the target switch, refer to the *LightStream 2020 Network Operations Guide*.

Step 4 Set the SNMP community to a read/write community by entering the following at the *cli> prompt:

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

where

<community name> = The SNMP read/write community that you want to access.

Step 5 At the *cli> prompt, enter:

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

where

- {enable|disable} Specifies whether the trap is enabled or disabled. The default is enabled.
- <trap#> = 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 (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 surround the expression in which it appears in quotes. For example, to specify all NDD traps, you would type **show trap "ndd*"**
- [<group name>] = An optional argument used to define a group of traps that you want to turn on or off. To use this argument, 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 Enter the following at the *cli> prompt to display the status of each trap in the MMA:

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

Expected Results

The disabled trap for the selected node is not passed to the CLI or displayed on a third-party NMS. The status display shows traps as either on, off, or disabled. If the status is either on or off, the trap is enabled. Otherwise the trap is disabled. See the section "Turning a Trap On or Off in a Specific Process" earlier in this chapter for a description of on and off.

Displaying 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 "Turning a Trap On or Off in a Specific Process" earlier in this chapter for a description of on/off. See the section "Enabling/Disabling a Trap for All Processes" earlier in this chapter for a description of enabled/disabled.

Procedure 1: View the Status of One or More Traps for a Process

Step 1 Verify that the target switch is correct by entering the following at the cli> prompt:

```
cli> show chassis general
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Network Operations Guide*.

Step 2 Set the SNMP community to a read/write community by entering the following at the `cli>` prompt:

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

where

`<community name>` = The SNMP read/write community that you want to access.

Step 3 At the `cli>` prompt, enter:

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

where

- `{<#>|<alias>} =` The number or the alias name of the process. See the section "Setting the Trap Reporting Threshold for Processes" earlier in this chapter for information on obtaining pid numbers and aliases.
- `<trap#>` = The number of the trap(s). Enter the symbolic trap name or trap number. You can set multiple traps 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 `*`, you must surround the expression in which it appears in quotes. For example, to specify all NDD traps, you would type **`show trap "ndd*"`**.
- `[<group name>]` = An optional argument used to define a group of traps on which you want to show status. To use this argument, 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.

Procedure 2: View the Status of One or More Traps for an MMA

Step 1 Verify that the target switch is correct by entering the following at the `cli>` prompt:

```
cli> show chassis general
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Network Operations Guide*.

Step 2 Set the SNMP community to a read/write community by entering the following at the `cli>` prompt:

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

where

`<community name>` = The SNMP read/write community that you want to access.

Step 3 At the `cli>` prompt, enter:

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

where

- `<trap#>` = The number of the trap(s). 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 `*`, you must surround the expression in which it appears in quotes. For example, to specify all NDD traps, you would type **`show trap "ndd*"`**.

- [<group name>] = An optional argument that defines a group of traps. To use this argument, 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.

Expected Results

If you enter **show trap ndd_3 ndd_4 ndd_5 ndd_1001**, the status of these traps is displayed, as shown in Figure 2-14.

Figure 2-14 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 **show trap "*"** the status of all traps in the MMA is shown in Figure 2-15. (Figure 2-15 is a partial display. Several screens of traps are displayed when you issue this command.)

Figure 2-15 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 two preceding procedures. If you do not create and maintain this file, you must manually enter each trap number used with those commands.

Follow the procedure below to create the cli.groups file.

Procedure to Create the cli.groups File

Step 1 To enter protected mode, enter the following at the `cli>` prompt:

```
cli> protected
```

Step 2 Enter the protected mode password when you see the following prompt:

```
Enter password:
```

Step 3 Escape from the CLI to the LynxOS bash shell by entering the following at the `*cli>` prompt:

```
*cli> shell bash
```

Step 4 Move to the /usr/app/base/config directory by entering the following at the prompt:

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

Step 5 Invoke the vi editor by entering the following at the prompt:

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

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

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

where

- `<groupname>` = A name that defines the group of traps.
- `<trap#>` = 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 **^[** followed by typing:

```
ZZ
```

Step 7 To return to the CLI, enter the following at the prompt:

```
LSnode:2# exit
```

Step 8 To exit protected mode, enter the following at the `*cli>` prompt:

```
*cli> exit
```