SNMP Managed Groups

The following is a list of Events management objects (SNMP Groups) that can be monitored or altered. This group of objects is defined in the AgentMon MIB.

SNMP Groups

This section lists the SNMP Groups and their test results

Group	Action
limits	OBJECT IDENTIFIER ::= { Events20 1 } This SNMP group is used to manage Events from a network management station. It allows the Network Manager to reconfigure Events in many of the same ways that the System Administrator can with the Configure Events GUI.
trapmanage	OBJECT INDENTIFIER::={ Events20 2 } This group is also used by the Network Manager. This group is used for enabling and disabling TRAPs and for changing the severity level associated with each TRAP.
os	OBJECT INDENTIFIER::={ Events20 3 } Displays read-only values for the operating system name, local time, and time of last reboot.
kernel	OBJECT INDENTIFIER::={ Events20 4} Displays values for various kernel-related statistics.These values are useful for a UNIX guru who likes to tweak the O/S for performance.
mbuf	OBJECT INDENTIFIER::={ Events20 5} Displays statistics for various network-related message buffers.
ncache	OBJECT INDENTIFIER::={ Events20 6} Displays statistics on the Name Cache (if present).
vm	OBJECT INDENTIFIER::={ Events20 7} Displays statistics about the virtual memory system.
cpu	OBJECT INDENTIFIER::={ Events20 8} Displays the CPU load (average growth of a run queue) and CPU time for the following states: kernel, user, system, and idle (as a percentage; total % of four states = 100%).

Group	Action
proc	OBJECT INDENTIFIER::={ Events20 9} Displays the number loaded, CPU time used, and the process size for a user-selected process. The process status is also available on some operating systems.
fs	OBJECT INDENTIFIER::={ Events20 10 } Displays a list of local filesystems, their size, and the amount of free space available. This may also include swap space.
files	OBJECT INDENTIFIER::={ Events20 11} For each user-specified file, this reports one or more of the user-specified items: filename, size, last accessed time, last time file was modified, or last string matched if 'file clamping' is selected and the user has specified one or more regular expressions.
qu	OBJECT INDENTIFIER::={ Events20 12} For each user-specified directory, this reports the total number of files and number of old files in the directory. This also has a variable field (for each directory) so the user can specify what 'old' is.
hardware	OBJECT INDENTIFIER::={ Events20 13} Displays a list of all hardware that was discovered on the system when Events was last started.
software	OBJECT INDENTIFIER::={ Events20 14} Displays a list of all software currently installed on the system, if such software was (de)installed with pkgadd. On SCO UNIX, the list also includes all software installed with custom.
printer	OBJECT INDENTIFIER::={ Events20 15} Displays the printer daemon's view of the current status for each printer being monitored.
disk	OBJECT INDENTIFIER::={ Events20 16 } This group is reserved for future use.

Group	Action
rpc	OBJECT INDENTIFIER::={ Events20 17} Displays RPC client and server statistics.
api1, ..., api6	OBJECT INDENTIFIER::={ Events 18-23 } This group allows Events to monitor user-provided data. If you have test scripts that already collect data, you could modify the scripts to write any data to a file. Then you could use one or more of the API groups to monitor the data in the file and check for High and Low thresholds. Like all other tests, a threshold violation could then send email or a TRAP, notify PEP, *and* be logged to the EMD. Logging of non-alarm data is also provided, just as it is with other tests. See Chapter 7, "Events," and Chapter 5, "Monitoring You System Network," in the *enlighten/DSM User Guide* for more details on using API tests.
usertraps	OBJECT INDENTIFIER::={ Events20 24 } This group shows the last TRAP PDU that was initiated through the EventsCli command line interface. The EventsCli is an "open" interface between your existing shell scripts (and other monitoring programs) and the Events alarm notification methods (PEP, TRAPs, EMD). This is best suited to users who are already monitoring site-specific data and have their own threshold logic. When your tests detect an alarm condition, they could call the EventsCli. This then gives you easy access to our event management framework. See "EventsCli (8N)" on page F-5 for more details.

MIB II Groups

This section lists the MIB II Groups and their object identifiers.

Group	Action
mib-2	OBJECT IDENTIFIER::={ mgmt 1 } Events20 also provides MIB-2 functionality in addition to UNIX MIB. MIB-2 primarily relates to statistics concerned with various network protocols.

Enterprise-Specific Traps

AgentMon can issue Enterprise-specific traps. Each trap is manageable via SNMP. A trap is issued whenever the corresponding measured value traverses an alarm threshold (if one is specified) and the particular trap is enabled. Our Internet Enterprise number is:

1.3.6.1.4.1.548.1.2.2

The first part "1.3.6.1.4.1.548" identifies "our" SNMP domain and the remaining numbers refer to a specific MIB of ours, in this case, our UNIX MIB: Events20.

To enable and disable traps, use the trapmanage group from your Network Management Application. Changes take effect immediately and become the new start-up values should Events be restarted later. Only Network Administrator can enable and disable Traps, via SNMP. If a new test is added to Events, and that test has an associated trap, the trap will be disabled.

For instance, the blocksFree trap refers to the amount of free space on a filesystem. There is a single trap type (trap #12), but you can enable and disable each "instance type" (i.e., enable/disable per filesystem). The Enterprise-specific traps and the reasons they are sent (alarm conditions) are defined in the following table.

Trap Name	Alarm Condition
1. mbufclfree	number of message buffer clusters
2. vmfree	amount of virtual memory remaining
3. swapfree	amount of free swap space
4. cpuload	average lenth of the CPU run queue
5. cpuuser	percentage of time CPU has spent in user mode
6. cpukernel	percentage of time CPU has spent in kernel mode
7. cpuwait	percentage of time CPU has spent in wait mode
8. procinstance	number of instances of a process has changed
9. procsize	size of a process has changed

Trap Name	Alarm Condition
10. proctime	CPU usage (time) of a process
11. procfree	number of free process slots in kernel
12. fsbfree	filesystem space (number of blocks free)
13. fsifree	filesystem inodes (number of inodes free)
14. filesize	the size of a file
15. filemtime	the modification time of a file
16. fileatime	the time of last access of a file
17. quedOldFiles	the number of old files in a directory
18. quedFiles	the total number of files in a directory
19. hwchange	a change in the hardware inventory
20. swchange	a change in the software inventory
21. printerState	a change in the status of a printer
22. api1trap	user-provided data
23. api2trap	user-provided data
24. api3trap	user-provided data
25. api4trap	user-provided data
26. api5trap	user-provided data
27. api6trap	user-provided data
28. fileclamp	a regular expression match occurred in a file
500. utrap500	a TRAP originating from an instance of the EventsCli
501. utrap501	a TRAP originating from an instance of the EventsCli
502. utrap502	a TRAP originating from an instance of the EventsCli

Trap Name	Alarm Condition
503. utrap503	a TRAP originating from an instance of the EventsCli
504. utrap504	a TRAP originating from an instance of the EventsCli
505. utrap505	a TRAP originating from an instance of the EventsCli
506. utrap506	a TRAP originating from an instance of the EventsCli
507. utrap507	a TRAP originating from an instance of the EventsCli
508. utrap508	a TRAP originating from an instance of the EventsCli
509. utrap509	a TRAP originating from an instance of the EventsCli
510. utrap510	a TRAP originating from an instance of the EventsCli
511. utrap511	a TRAP originating from an instance of the EventsCli