SMPTE REGISTERED DISCLOSURE DOCUMENT

RDD 1-2003

DDRMON-MIB-1 — A Digital Disk Recorder Simple Network Management Protocol Management Information Base

The attached document is a Registered Disclosure Document prepared by the proponent identified below. It has been examined by the appropriate SMPTE Technology Committee and is believed to contain adequate information to satisfy the objectives defined in the Scope, and to be technically consistent.

This document is NOT a Standard, Recommended Practice or Engineering Guideline, and does NOT imply a finding or representation of the Society.

Errors in this document should be reported to the proponent identified below, with a copy to eng@smpte.org.

All other inquiries in respect of this document, including inquiries as to intellectual property requirements that may be attached to use of the disclosed technology, should be addressed to the proponent identified below.

Proponent contact information:

Northon Rodrigues

Thomson Broadcast and Media solutions
15655 SW Greystone Ct.
Beaverton OR 97006 (USA)

Email Northon.Rodrigues@Thomson.net

Approved
July 28, 2003

1 Scope

This Registered Disclosure describes a Management Information Base (MIB) for a digital disk recorder. A MIB, when used with the Simple Network Management Protocol (SNMP), allows a remote computer to monitor the status, set certain device management variables and receive notifications when certain conditions occur.

Additional MIBs may be used to monitor and manage other digital disk recorders and additional types of broadcast devices. This MIB does not describe a general-purpose device control protocol.

2 Normative references

The following documents contain provisions, which, through reference, constitute provisions of this document:

RFC 1155, Rose, M. and McCloghrie, K., “Structure and Identification of Management Information for TCP/IP-based Internets”, May 1990

RFC 1157, Case, J., Fedor, M., Schoffstall, M. and Davin, J., “A Simple Network Management Protocol”, May 1990

RFC 1212, Rose, M. and McCloghrie, K., “Concise MIB Definitions”, March 1991

RFC 1902, Rose, M. and McCloghrie, K., “Structure of Management Information for Version 2 of the Simple Network Management Protocol”, January 1996

RFC 1903, Rose, M., Case, J., Waldbusser, S., and McCloghrie, K., “Textual Conventions for Version 2 of the Simple Network Management Protocol”, January 1996

RFC 1904, Rose, M. and McCloghrie, K. “Conformance Statements for Version 2 of the Simple Network Management Protocol”, January 1996

RFC 1905, Rose, M., Case, J., Waldbusser, S., and McCloghrie, K., ”Protocol Operations for Version 2 of the Simple Network Management Protocol’, January 1996

RFC 1907, Rose, M. and McCloghrie, K., “Management Information Base for Version 2 of the Simple Network Management Protocol”, January 1996

3 Definitions

3.1 agent: An entity that assumes the operational role of receiving, processing, and responding to requests, and generating event reports. It must have access to the network management information to respond to requests, and must be notified of internal events to generate reports.

3.2 columnar or table object: An object that has zero, one, or more instances at any point in time. For instance, the state of a TCP connection is a columnar object. There may be zero, one, or more TCP connections to a system at any point in time. Each TCP connection that exists has a state.

3.3 conceptual MIB table: The SMI requires related columnar objects to be organized into conceptual tables. One or more columnar objects defined in the same table or another table uniquely identify the instance of each columnar object in a table. These objects are collectively referred to as the table’s index variables.

3.4 data types: Table 1 describes the data types or the syntax of the objects discussed in this document.

Table 1 – Data type descriptions

Data Type	Description
INTEGER	Enumerated 32-bit signed integer whose value varies between (-2^31) and (2^31-1)
Integer32	32-bit signed integer whose value varies between (-2^31) and (2^31-1)
OCTET STRING	Eight-bit bytes of binary or textual information. The maximum number of octets in an object's value is limited to 65535 (2^16-1) octets.
IpAddress	A string of four octets. A value of this type is an IPv4 address encoded in network byte order
Counter32	A non-negative 32-bit integer whose range includes only non-negative integers. The reported value for objects of this type is a count, modulo 2^32, of the occurrence of an event or the measure of flow. The starting value of a counter is undefined unless specified where the type is used.
Gauge32	Specify a value whose range includes only non-negative integers. The values range from zero to (2^32-1). These types have the behavior such that the reported value is always within the specified range even when the actual value is outside the specified range.
DisplayString	A stream of eight-bit bytes where each byte is a character from the NVT-ASCII character set. Note that this stream is not null-terminated.
TimeTicks	A non-negative value, whose range includes only non-negative integers. The reported value is an elapsed time, modulo 2^32, in units of hundredths of a second between two events (or epochs). The length of time between rollovers is approximately 497 days.
PhysAddress	A stream of eight-bit bytes where the collective stream of bytes defines the data-link layer address of a network interface.

Numerical values in this document that are preceded by 0x are hexadecimal values. All other numerical values are decimal numbers.

3.5 management information base (MIB): Definitions of related management information, events, and associated implementation compliance requirements are specified in documents called management information base (MIB) specifications. These specifications include prose sections and computer readable descriptions. The MIB modules are written in an adapted sub-set of the ASN.1 language. In essence, MIBs are specifications containing definitions of management information so that networked systems can be remotely monitored, configured, and controlled.

3.6 manager, or management station: An entity that assumes the operational role of generating requests to retrieve or modify management information, receiving responses to requests, or receiving event reports. This role is assumed by (or on behalf of) a management application.

3.7 objects: In the SNMP framework, a system is managed by retrieval and modification of management information. Each class (or type) of management information is called an object. The definition of an object

type includes its data type, the maximum allowed access, its assigned identity, how instances are identified, and its semantics (or behaviors).

3.8 object group: A collection of logically related objects within a MIB module. It refers to a collection of scalar objects and/or columnar and scalar objects. Grouping objects aids in maintenance and improves organization of the MIB module. Groups also permit selective implementation of related managed objects on the agent.

3.9 object identifier (OID): An identification scheme conceptualized in abstract syntax notation one (ASN.1) is used to uniquely identify managed objects by assigning an identifier to each object. This identifier is administratively assigned such that it uniquely identifies the object for all space and time and is called an object identifier (OID). An OID value is an ordered sequence of non-negative integers, written from left to right, containing at least two elements. SNMP imposes a restriction on the maximum length of any OID value to 128 numbers in the sequence and the maximum value of a number to 2^32-1. Each number in the sequence is called a sub-identifier in the SMI. An OID value is usually written as a sequence of numbers separated by a dot (.) and optionally labeled.

3.10 object instance or variable: A specific instance from a class of management information is called a variable or an object instance.

3.11 scalar objects: An object that has exactly one instance or value. An example of a scalar object is the location of a managed system, since a system has exactly one location.

3.12 SnmpGet: An SNMP operation used to retrieve the value of an object's instance. The operand is a list of identities, which exactly match the identity of returned variables.

3.13 SnmpGetNext: An SNMP operation used to retrieve the value of an object's instance. The operand is a list of identities, which are approximations of the identity of returned variables. Each returned identity is the one assigned to the first accessible variable whose identity is greater than the given identity.

3.14 SnmpSet: An SNMP operation used to modify the value of an object's instance. The operand is a list of pairs. Each pair consists of the identity of a variable and its desired value. SnmpSet operations are used to configure and control a managed system.

3.15 SnmpTrap, or SnmpNotification: An SNMP event reporting operation, which specifies an event and zero or more list of pairs. A pair consists of the identity of a variable and its value. This operation is used to report the occurrence of events on a managed system to a list of managers configured to receive events for that managed system.

3.16 structure of management information (SMI): The rules for writing SNMP MIBs defined using a subset of ASN.1. These rules include the type definition or syntax of objects, rules to define objects, notifications, and other constructs used in the MIBs. RFC1155 defines the SMI for version 1. RFC1212 called the SNMPv2-SMI supersedes RFC1155-SMI and enhances it. This is the definition of SMI version 2. The DDRMON-MIB1 MIB uses SMIv2 for its MIB definitions.

3.17 VITC: Vertical interval time code.

4 DDRMON-MIB1 MIB overview

The DDRMON-MIB1 MIB is a specific MIB module that contains SMIv2 compliant definitions of managed objects used for networked monitoring of digital disk recorders.

The MIB is classified into four major groups:

- The *pvsNotifications* group which defines all the DDRMON-MIB1specific notifications

- The *pvsGeneral* group which defines generic diagnostic objects and objects that allow fine tuning of the DDRMON-MIB1's internal monitoring engine.

- The *pvsCardCage* group which defines the slot-map information of the disk recorder card cage and board specific status information.

- The *pvsSubSystems* group which further segregates object definitions depending on logical subsystems on the disk recorder.

The MIB objects descriptions discussed in the document use the following conventions:

- Default integer enumeration: *In Italic.*

- Objects that can accept a SnmpSet operation (Read/Write objects): *In Italic.*

- Objects that are a table index: **In Bold**.

5 *pvsGeneral g*roup

This group defines objects which aid in diagnostics, as well as a group dedicated to the configuration of the fault indicators incorporated into the disk recorder.

5.1 General system level diagnostic aiding objects

These objects, although not directly initiating or reporting the results of system diagnostics, provide general reporting which may assist in interpreting the results of diagnostics.

Table 2 – Generic diagnostic objects

Object	Syntax	Usage
PvsMode	INTEGER { *normal-mode(1),* maintenance-mode(2) }	The disk recorder system software runs in two distinct modes. In normal-mode, media operations can be performed the system. In maintenance-mode, only maintenance operations such as running system diagnostics can be performed.
pvsLastTrapCategory	INTEGER { alarm(1), warning(2), informational(3), resent-alarm(32769), resent-warning(32770) }	This object holds the category of the trap that was last dispatched. It can be used by a manager to decode the categorization of disk recorder MIB traps and is included as the first enterprise variable in every disk recorder MIB trap. Resent-alarm and resent-warning used when the system resends previously sent warnings and alarms that are not handled or acknowledged within a stipulated time.
pvsSerialCode	DisplayString	The serial number assigned to this disk recorder.
pvsSoftwareBuild	DisplayString	The disk recorder software build number.
pvsModel	DisplayString	Indicates the disk recorder model.
pvsVerLow	Integer32	The lower order version-stamp of the SNMP software that is used on the disk recorder. This number matches the number of revisions to this MIB module.
pvsVerHigh	Integer32	The higher order version stamp of the SNMP software that is used on the disk recorder. This number implies major upgrades to the agent software, for instance support for new MIB modules.

5.2 Fault indicator configuration group

Along with SNMP-based monitoring, the disk recorder uses public and private logs to trace all its operations. A fault indicator installed on the front panel indicates operational anomalies, and a subset of the GPI output ports is used for general purpose electrical switching to trigger customized hardware-based notifications. The objects in this group provide control and configuration of the behavior of these fault indicator resources.

Table 3 – Fault indicator objects

Object	Syntax	Usage
pvsFaultLed	INTEGER { *led-off (1),* led-on (2) }	The system provides a general-purpose front panel indicator (LED) which signals system status. An SnmpSet operation with the value 'led-on(1)' lights the indicator, signaling a system fault.
pvsTrapResendEnable	INTEGER { trap-resend-disable(1), *trap-resend-enable (2)* }	The system can resend SNMP warning or alarm traps periodically until it detects a normal condition for that alarm or warning. This variable disables or enables the trap resend. By default, resend is enabled. A change to this value takes effect only after a system reboot, although a subsequent SnmpGet prior to the reboot returns the new value.
pvsTrapResendInterval	TimeTicks	This variable holds the warning or alarm trap resend frequency in hundredths of a second. The default value is 360000, which corresponds to a default resend interval of one hour. A change to this value takes effect only after a system reboot, although a subsequent SnmpGet prior to the reboot returns the new value.
PvsSuppressionInterval	TimeTicks	Under certain circumstances. the system may log the same information repetitively. This variable suppresses repeated messages for the specified duration. A value of 0 indicates that the system will not suppress any repeated messages. The default value is 180000, which i suppresses repeated log messages for a duration of half an hour. A change to this value takes effect only after a system reboot, although a subsequent SnmpGet prior to the reboot returns the new value.

The GPI output control and configuration objects are organized into a table where each row of the table depicts a physical GPI output port on the disk recorder. This table is indexed on the GPI port number. Typically the rows in this table will only include entries for those GPI output ports which are specifically reserved for fault indication

Table 4 – GPI output control objects

Object	Syntax	Usage
PvsGpiOp	Integer32	A unique value that identifies each physical GPI output port. This is the index variable for pvsGpiOpPortTable.
PvsGpiOpActiveLevel	INTEGER { active-low (1), *active-high (2)* }	This variable holds the digital level that the GPI port produces when it is triggered. This value is valid only if the corresponding port value for pvsGpiOpState is not 'disabled(1)'. A change to this value takes effect only after a system reboot, although a subsequent SnmpGet prior to the reboot returns the new value.
PvsGpiOpTriggerEvent	INTEGER { *alarm (1),* warning (2), alarm-and-warning(3) }	The category of traps generated by the system that trigger the GPI port. This value is valid only if the corresponding value for pvsGpiOpState is not 'disabled(1)'. A change to this value takes effect only after a system reboot, although a subsequent SnmpGet prior to the reboot returns the new value.
PvsGpiOpState	INTEGER { *disabled(1),* enabled (2), activate (3), triggered(4), deactivate(5) }	The state and control object for a GPI output port. By default, its value is 'disabled(1)' to indicate that the port is not used. If an SnmpSet operation is performed on this object to set its value to 'disabled(1)' when the port is in the 'triggered(3)' state, the port is deactivated and the settings of other columns in the table are restored to their default factory-defined states. However, this GPI port cannot be re-assigned for other uses until the system undergoes a cold reboot (power off, then power on). If an SnmpSet operation is performed on this object to set its value to ‘enabled(2)' when the port is 'disabled(1)', the change takes effect after a system reboot, even though subsequent SnmpGet operations return 'enabled(2)', This prevents acquisition of the port while it is allocated for another purpose. When a SnmpSet operation sets this object to ‘enabled(2)', the system registers other values of other columns in this table associated with the specified index. NOTE: If an SnmpSet operation sets this object to 'enable(2)' with a column index 'i' and the corresponding value of this variable for another lexicographically greater index 'j' is 'disabled(1)', then the set operation fails with the error code ‘badValue’. The system can only allocate GPI ports for error notifications from the last port onwards. The operation succeeds only if the value of this variable for the lexicographically greater index 'j' was previously set to 'enable(2)'. If the system sets the value of this variable as 'enabled(2)' following a cold boot, it attempts to auto-activate the port on all fault conditions of the configured trigger event category. Similarly, when the system detects no pending faults of the trigger event category, it attempts to auto-deactivate the port.

		However, this variable also provides an option to manually activate and deactivate the port. To activate the external equipment connected to the port this variable should be set to a value of 'activate(3)'. A value of 'activate(3)' can be set only if the previous value was 'enabled(2)' else the set operation will return an error code of badValue. This value is used only for activating the port and will never be returned in response to a SnmpGet operation. After a successful activation, if an SnmpGet operation is performed, a value of 'triggered(4)' will be returned to indicate that the external device connected to the port is triggered. This value is meant only to indicate the current state of the port and an SnmpSet operation with this value has no effect whatsoever on the operation of the port, or the other column variables. To reset the external equipment to the non-triggered state after it has been triggered, a set should be performed with the value of 'deactivate(5)'. An SnmpSet operation with this value is accepted only if the port was in the 'triggered(4)' state. If the previous value was 'disabled(1)' the set operation returns an error code of badValue. This value is used only for deactivating the port and is never returned in response to an SnmpGet operation.

6 Board status objects: *pvsCardCage* group

The objects in this group include a variable *pvsCcNumSlots* to indicate the maximum possible board slot configuration on the disk recorder and the *pvsCcBoardTable,* which enumerates the slot map on the disk recorder and the individual status of each installed board. None of the boards described by the objects in this table are hot-swappable. Each entry in the table is uniquely identified by the slot number in which the board represented by the table row is installed, and the table does not contain any row depicting an empty slot.

Table 5 – Objects in the *pvsCcBoardTable* table

Object	Syntax	Usage
PvsCcBoardSlot	Integer32	A unique value for each installed board. Its value ranges between 1 and pvsCcNumSlots. It depicts the slot location number. The count starts from the Applications Processor Board side of the card cage. This table does not enumerate the Mother Board which holds the slot array, nor does it enumerate the Applications Processor Board.
PvsCcBoardType	INTEGER { refgen[1] (1), sdi4out[2] (2), sdi2in2out[3] (3), audio[4] (4), smpte314m-codec[5] (5) , sdti[6] (6) , mpeg2-encoder[7] (7), mpeg2-decoder[8] (8) , fibre-channel-disk[9] (9), video-monitor[10] (10) , hd-sdi[11] (11), hd-mpeg2-encoder[12] (12) , real-time-system[13] (13), hd-mpeg2-decoder[14] (14) , video-processor-4codec[15](15),	The possible types of boards that can be plugged into the slot array on the motherboard.

[1] Reference signal board - SMPTE 318M
[2] Four output SDI board - SMPTE 259M & SMPTE 272M-1994
[3] Two input and two output SDI board - SMPTE 259M
[4] Audio I/O board - AES 3-1997
[5] DV family video compression board - SMPTE 314M.
[6] SDTI interface board - SMPTE 305M
[7] Dual MPEG-2 encoder board - ISO 13818-2
[8] Dual MPEG-2 decoder board - ISO 13818-2
[9] Fibre Channel disk interface board - ANSI X3.230, and ANSI X3.297
[10] Four channel video monitoring board - SMPTE 170M & ITU (CCIR) 624-4
[11] HD SDI board - SMPTE 292M & SMPTE 299M-1997
[12] HD MPEG-2 encoder board - ISO 19818-2
[13] Real-time systems control board.
[14] HD-MPEG-2 decoder - ISO 13818-2
[15] Video processor board.
[16] SD composite video board - SMPTE 170M and ITU-R (CCIR) Report 624-4:1990

	composite-video[16] (17) , real-time-system-ii[17] (18), video-processor-2codec[18] (19) , rs422[19] (241), fibre-channel-network[20] (242) , ethernet[21] (243) , unknown (255) }	
PvsCcBoardRev	DisplayString	The hardware revision-stamp of a board. This is a string in x.yz format where x is the board revision number and yz is the board module increment.
PvsCcBoardStatus	INTEGER { ok (1), dead (2), generic-error (3), unknown-error (4), in-maintenance (5) , communication-error (6) , sdiDsp1Error (1001), sdiDsp2Error (1002) , sdiVitcReadError (1003), sdiVitcCrcError (1004) , sdiAudioSyncError (1005), sdiAudioExtractError (1006) , videoStorageFcPort1Failed (9001), videoStorageFcPort2Failed (9002) , videoStorageFcPortsFailed (9003), videoStorageBandwidthOverflow (9004) , videoStorageScsiError (9005), videoStorageAccessible (9006) , videoStorageLoopFailover (9007), videoStorageInaccessible (9008) ,	The operational status of a board. The errors are categorized based on the type of the board. Specific board status values depend on the type of board installed in the slot. **Generic Board Status:** These values apply to any installed board. ok(1) – The board is operational. dead(2) - The system has inferred that the processor on this board is dead and not responding. not-present(3) - A previously detected board is missing. unknown-error(4) – An unidentified error is reported by a board. in-maintenance(5) - The board is operational, but not running in a normal mode. This mode is typically reported in instances when the diagnostics are being run on the board. communication-error(6) - The processor on the board may be alive, but is unable to communicate with the processors on the other boards. **Serial Digital Interface Board Status:** These values may occur if the board installed in this slot is an sdi4out or sdi2in2out. sdiDsp'n'Error(1001, 1002) - The DSP processor 'n' on the board has been detected to be dead. Video input or output processing on the channels associated with this board are affected.

[17] Real-time systems control board (version II).
[18] Video processor board (version II).
[19] Eight port RS-422 interface board - ANSI/TIA/EIA-422-B
[20] Fibre Channel data network board - ANSI X3.230, and ANSI X3.297
[21] Ethernet 10/1000 BaseT data network board - IEEE 802.3

		associated with this board are affected. sdiVitcReadError(1003) - An error was encountered while extracting VITC data from video on the input channels associated with this board. There could be an error in video input cabling or the input video feed. sdiVitcCrcError(1004) – Checksum errors were encountered while interpreting VITC packets from video on the input channels associated with this board. sdiAudioSyncError(1005) – The embedded audio extracted from video on the input channels associated with this board was repeatedly observed to be out of sync with video. SdiAudioExtractError(1006) - Errors were encountered while extracting embedded audio packets from video on the input channels associated with this board. **Fibre Channel Disk Board Status:** VideoStorageFcPort'n'Failed(9001, 9002, 9003) – These conditions are reported when a hardware defect is detected on either of the Fibre Channel ports through which the system accesses the RAID array. VideoStorageBandwidthOverflow(9004) - The configured media compression bit-rate exceeded the maximum possible limit. Media capture tasks may have failed. VideoStorageScsiError(9005) - An abnormal SCSI bus or disk drive event was detected and could not be cleared by the RAID controller through retry attempts. VideoStorageAccessible(9006) - The disk recorder has established communication with the external storage equipment. VideoStorageLoopFailover(9007) - The disk recorder was unable to communicate with the external storage equipment over the current Fibre Channel arbitrated loop and has switched over to the redundant (backup) loop. videoStorageInaccessible(9008) – The disk recorder was unable to access the external storage and is unusable. **Video Network Board Status:** videoNetworkNotConnected(242001) - The network adapter on the disk recorder has lost connectivity with the Fiber Channel hub or switch. videoNetworkConnected(242002) - The network adapter on the disk recorder has established connectivity with the Fiber Channel hub or switch. videoNetworkAdapterDead(242003) - The network adapter on the disk recorder has stopped responding and video transfers fail. This typically occurs when the disk recorder initializes its Fibre
	videoNetworkNotConnected (242001), videoNetworkConnected (242002) , videoNetworkAdapterDead (242003), mpeg2Decoder1Failed (8001), mpeg2Decoder2Failed (8002) , mpeg2DecoderIpmError (8003), mpeg2DecoderRsrcReqFailed (8004), mpeg2DecoderDiskAccessFailed (8005) , mpeg2DecoderBufferOverflow (8006), mpeg2Encoder1Failed (7001), mpeg2Encoder2Failed (7002), mpeg2EncoderIpmError (7003), mpeg2EncoderRsrcReqFailed (7004), mpeg2EncoderMediaDriverError (7005) , mpeg2EncoderHardwareFailure (7006), mpeg2Encoder1BadDiskResponse (7007), mpeg2Encoder2BadDiskResponse (7008) , mpeg2Encoder1DiskResponseOk (7009), mpeg2Encoder2DiskResponseOk (7010), audioPacIpmStartupError (4013) , audioPacSioIntError (4014), audioPacTtyCreateError (4015) , audioPacTtyOpenError (4016), audioPacInvalidMsg (4017) , audioPacIpmWriteError (4018), audioPacResetFailed (4019) , audioPacReset (4020), audioPacLostContact (4021), audioPacRegainedContact (4022) ,	

	audioPac208Detected (4023), audioPac216Detected (4024), audioXlr216Detected (4025), rtpOutOfSync (10001), rtpInputLocked (10002) , rtpInitFailed (10004),	Channel network adapter that is not connected to the network. **MPEG-2 Decoder Board Status:** mpeg2Decoder'n'Failed(8001, 8002) - Decoders 1 or 2 failed to initialize. mpeg2DecoderCommError(8003) – The decoder board driver is not able to establish inter-processor communication at startup. mpeg2DecoderRsrcReqFailed(8004) - The board failed to initialize because it was deprived of a system-wide resource such as a DMA channel. mpeg2DecoderDiskAccessFailed(8005 - The decoder board encountered an error while accessing the storage system. mpeg2DecoderBufferOverflow(8006) - The decoder board is unable to decode due to insufficient memory. **MPEG-2 Encoder Board Status:** mpeg2Encoder'n'Failed(7001, 7002) - Encoders 1 or 2 failed to initialize. Mpeg2EncoderIpmError(7003) - The encoder board driver is not able to establish inter-processor communication at startup. Mpeg2EncoderRsrcReqFailed(7004) - The board failed to initialize because it was deprived of a system-wide resource such as a DMA channel. mpeg2EncoderMediaDriverError(7005) - The system is unable to communicate with the encoder board due to a communication error with the encoder's media driver. mpeg2EncoderHardwareFailure(7006) - A generic hardware failure on the encoder board. mpeg2Encoder'n'BadDiskResponse(7007, 7008) - The storage system is responding too slowly to the record task on the encoder 'n'. Video may be dropped. mpeg2Encoder'n'DiskResponseOk(7009, 7010) - The storage system is responding normally to the record task on encoder 'n', after a period when it responded too slowly. **Audio Board Status:** AudioPacComStartupError(4013) - The audio board was unable to establish inter-processor communication for controlling the external audio option. AudioPacSioIntError(4014) - The audio board was unable to connect to the external audio option's SIO interrupt. AudioPacTtyCreateError(4015) - The audio board was unable to create a tty for the external audio option. AudioPacTtyOpenError(4016) - The audio board was

		unable to open the tty for the external audio option. AudioPacInvalidMsg(4017) - The audio software received an invalid inter-processor communication message during startup or configuration. AudioPacIpmWriteError(4018) - The audio software encountered an inter-processor messaging write error during startup or configuration. audioResetFailed(4019) - The external audio option failed to reset. audioPacReset(4020) - There was a temporary loss of contact between the audio board and the external audio option, and the board has reset to the last known good state. This results in an audio glitch. audioPacLostContact(4021) - The audio board has lost contact with the external audio option, and the system attempts to regain contact. AudioPacRegainedContact(4022) - The audio board has regained connectivity with the external audio option. audioPac208Detected(4023) - The audio board has detected a non-supported external audio option, the PAC208 unit. This will result in unpredictable results. audioPac216Detected(4024) - The audio board has detected a PAC216 external audio option. audioXlr216Detected(4025) - The audio board has detected an XLR216 unit. **Real-Time-System Board Status:** RtpOutOfSync(10001) – The reference timing system, that was locked into the reference black (house black) signal, has gone out of sync. rtpInputLocked(10002) – The reference timing system has locked onto the reference (house) black signal. rtpInitFailed(10004) – The board did not initialize.

7 Subsystem grouping: *pvsSubSystems*

The disk recorder monitoring system breaks up the functional complexity of the entire system into discrete logical subsystems, a failure threat or fault to any of which would cause concern to the system's overall operation. These subsystems may exist on or as resources on the same physical board, or may be spread across multiple boards. Moreover, multiple subsystems could exist on the same board or hardware resource, or could be completely external to the mechanical composition of the disk recorder.

7.1 Storage subsystem: *pvsVideoStorage*

The storage subsystem defines the objects that describe the particulars of the disk recorder's external RAID storage equipment and the media file system that resides on the external storage. The RAID objects proxy the Fibre Channel SCSI RAID information. If the disk recorder is not connected, or has a bad communication link with the external storage, the disk recorder is unable to furnish the correct values for the RAID group objects.

7.1.1 RAID group: pvsVsRaid

The RAID group divides the RAID information into three tables, each identifying information about a major class of components in the external RAID storage assembly: the RAID controller table, the RAID chassis table, and the RAID drives table. The disk recorder defines a highly scalable and redundant RAID equipment architecture.

Due to the multiple existence and inherent design of the RAID system, each RAID chassis is identified not only by the chassis address, but also by the address of the RAID controller that conveys information about a particular component to the disk recorder. Similarly, each drive module in the drive table is identified by the drive slot location within a particular chassis and the identification information for that chassis which is the address of the controller controlling all units in that chassis and the address of the chassis. However, to avoid redundant index objects in all the tables, the controller address is defined in the controller table, the chassis address is defined in the chassis table, the drive slot location in the drive table and the table indices refer to these objects appropriately. For instance, a row in the drive table refers to the controller address defined in the controller table, the chassis address in the chassis table and the drive slot number in the drive table.

To maintain consistency with the notion that integer-based indices for a MIB table cannot be zero-based in order to distinguish these objects from the scalar objects, all the RAID components IDs are one-based rather than zero-based, as they are on the actual hardware. When interpreting the information, these IDs should be converted to zero-based.

Table 6 – RAID controller table

Object	Syntax	Usage
pvsControllerId	Integer32	A unique value identifying an external RAID controller. This is the Fibre Channel arbitrated loop address set on the controller board using the Fibre Channel arbitrated loop ID switches. The Fibre Channel arbitrated loop protocol translates the ID into an 8-bit arbitrated loop physical address. NOTE: Although the value of this variable is supposed to indicate the zero-based number as set on a RAID controller board, it is a one-based number in the MIB. When locating a physical controller board, this value must be converted to a zero-based number. For example, if the controller board number is 'i', then the corresponding value of pvsControllerId for that controller in the MIB is 'i+1'.
pvsControllerStatus	INTEGER { spOk (1), spAdapterFailure (2), spAdapterPanic (3), spCacheDisabled (4), spCacheMemoryFault (5), spVaultLoadError (6), spNvRamError (7), spDbaseSyncError (8), spPeerControlError (9) }	The operational status of each RAID controller accessed by the disk recorder. spOk(1) - The RAID controller card is operating under normal conditions. spAdapterFailure(2) - The RAID controller card has failed. spAdapterPanic(3) - The RAID controller's firmware encountered a situation from which there was no reliable execution other than reloading itself from disk and restarting. spCacheDisabled(4) - Some hardware condition has caused the controller to disable the cache. spCacheMemoryFault(5) – The controller (operating in non-mirrored caching mode) has detected a non-recoverable memory fault in the cache memory area.

		spVaultLoadError(6) - The controller encountered errors in loading the cache image from the disk, which may indicate multiple drive failures. spNvRamError(7) - The controller's NV-RAM was not initialized, and the controller has initialized it to the default state. The controller will be functional after a reset. spDbaseSyncError(8) - The controller is unable to determine the correct logical configuration of all the units in the system. This can result in some units being unusable. spPeerControlError(9) – The controller's peer controller is no longer responding to queries from this controller (after the controller detected the presence of the peer controller).
PvsControllerVideoFlareVer	DisplayString	The version of the Video Flare (the operating system) running on this RAID controller. The version number string appears as x.y.z, where x is the major version, y is the minor version and z is the pass revision number.

Table 7 – RAID chassis table

Object	Syntax	Usage
pvsChassisId	Integer32	A unique value for each RAID chassis associated with a RAID controller that determines the disk module address. A RAID chassis that contains a RAID controller is always assigned the value 0. This is assigned using the selector on the front panel of the chassis. NOTE: Although the value of this variable is supposed to indicate the zero-based number set on the chassis, it is a one-based number in the MIB. When locating a physical chassis, this value must be converted to a zero-based number. For example, if the number set on the chassis is 'i', then the corresponding value of pvsChassisId for that chassis is 'i+1' in the MIB.
pvsChassisNumDisks	Integer32	The number of disk drive modules installed in this chassis. This can be either 5 or 10 (1 or 2 logical units, where each logical disk unit comprises 4 data drives accompanied by a single data parity drive).
pvsChassisNumControllers	Integer32 (0..2)	The number of RAID controllers in this chassis. The disks in a chassis might controlled by the controller board in another chassis, A given chassis may contain up to two controller boards to control the disk modules.
pvsChassisNumLccs	Integer32 (0..2)	The number of link control cards used to daisy chain several SCSI RAID chassis.

pvsChassisStatus	INTEGER { daeOk (1), daeFanDisabled (2), daeFanShutdown (3), daeFanOk (4), daeOverheating (5), daeTemperatureOvershoot (6), daeStandByPowerRemoved (7), daeStandByPowerFailed (8), daeLinkAdapterError (9) }	The operational status of the external RAID chassis as reported by the active controller in the chassis. **Chassis Fan Status:** DaeFanDisabled(2) - The fans in the chassis were disabled. The controller shuts down the drives in the chassis and transitions to standby mode if the fans are not operational within the next two minutes. DaeFanShutdown(3) – The drive fan pack was shut down or removed. daeFanOk(4) - The fan operation on the chassis has been restored. **Chassis Temperature Indications:** daeOverheating(5) - The chassis is overheating. The controller attempts a corrective action. DaeTemperatureOvershoot(6) - The temperature overshoot in the chassis cannot be corrected. All the drives in the chassis will be shut down. **Chassis Power Supply Indicators:** DaeStandByPowerRemoved(7) - The standby power supply unit has been removed. daeStandByPowerFailed(8) - The standby power supply unit has failed. The system disables write caching. **Chassis Link Adapter Status:** daeLinkAdapterError(9) - The link control card on the chassis has failed.
pvsChassisModel	DisplayString	The RAID chassis model.

Table 8 – RAID drives table

Object	Syntax	Usage
pvsDriveId	Integer32 (1..10)	A unique value for each media drive module. This identifies the slot location of the drive module in the chassis. Each physical drive slot on the chassis is identified. This value ranges between 1 and pvsChassisNumDisks. NOTE: Although the value of this variable is supposed to indicate the zero-based number as labeled on the RAID chassis, it is a one-based number in the MIB. When locating a physical drive within a chassis, this value must be converted to a zero-based number. For example, if the drive number on the chassis is 'i', then the corresponding value of pvsDriveId for that drive is 'i+1' in the MIB.
pvsDriveStatus	INTEGER { driveOk (1), driveRebuilding (2), driveRebuildCompleted (3), driveRebuildAborted (4), driveShutdown (5),	The status of each media drive-module. driveRebuilding(2) - The RAID controller is rebuilding data on the drive. driveRebuildCompleted(3) - The RAID controller has completed rebuilding data on the drive. driveRebuildAborted(4) - The RAID controller has

	driveReplace (6), driveFailure (7), driveReplaceError (8), driveUnsupportedSoftware (9) , driveUnsupportedHardware (10), driveUnsupported (11), driveSoftMediaError (12), driveMediaError(13), driveHardScsiBusError (14) }	aborted a rebuild on the drive. driveShutdown(5) - The drive module was shut down. driveReplace(6) - The drive module is susceptible to an imminent failure and should be replaced. driveFailure(7) - The drive module has failed and cannot be accessed. driveReplaceError(8) - The drive module was replaced, but the replaced module does not have the capacity to support a rebuild. driveUnsupportedSoftware(9) - The controller software does not support this drive module. The module is rendered useless and should be replaced. driveUnsupportedFirmware(10) - The controller software does not support the drive module's firmware. The module must be replaced. driveUnsupported(11) - The system does not support this module. driveSoftMediaError(12) – The drive module reported a defect that was successfully cleared by the controller. If these errors are reported frequently, a module replacement is recommended. DriveHardMediaError(13) - The drive module reported a media defect that could not be cleared by the controller. DriveHardScsiBusError(14) - The drive module reported a hard SCSI bus error. The drive could be inaccessible.
pvsDriveRebuildStatus	Integer32 (0..100)	Indicates a drive module rebuild progress. This holds a non-zero percentage only when pvsDriveStatus indicates the 'driveRebuilding' status. In other cases this always returns a zero value.
pvsDriveMicrocodeVer	DisplayString	The RAID drive module microcode revision number.
pvsDriveModel	DisplayString	The RAID drive module model.
pvsDriveSerialCode	DisplayString	The RAID drive module serial number.

The RAID group also defines the following two scalar variables:

Table 9 – Scalar objects in the pvsRaid group

Object	Syntax	Usage
pvsRdChassis	Integer32	The total number of external RAID chassis connected to the disk recorder.
pvsRdFibreChannelLoopStatus	INTEGER { active-loop-A (1), active-loop-B (2), not-connected (3) }	Indicates the connectivity status between the disk recorder and the external storage. active-loop-A(1) - indicates that the disk recorder is using the primary Fibre Channel arbitrated loop A to communicate with the external storage. active-loop-B(2) – indicates that the disk recorder is using the redundant Fibre Channel arbitrated loop B

		to communicate with the external storage. This could also indicate a failure on loop A that caused a fail-over to loop B. not-connected(3) – the disk recorder is not able to communicate with the RAID controllers and storage cannot be accessed. This may be caused by a loss of physical connectivity between the disk recorder and the external storage equipment.

7.1.2 Media file system group: pvsVsFileSystem

The disk recorder uses a single-volume media file system for video storage. The group enumerates the basic attributes of this volume.

Table 10 – Media file system objects

Object	Syntax	Usage
pvsFsBlockSize	Integer32	The granularity of the blocks on the media file system in bytes. The default value of this variable is 4096.
pvsFsFreeBlocks	Gauge32	The number of free blocks on the media file system. The size in bytes of each block is available as the value of pvsFsBlockSize.
pvsFsUsedBlocks	Gauge32	The number of blocks on the media file system allocated for media storage. The size of each block in bytes is available as the value of pvsFsBlockSize.
pvsFsDataSetAlias	DisplayString	The file system data set alias or file system volume name.
pvsFsCapacityThreshold	Integer32 (80..95)	The system sends an SNMP trap when the file system allocated space crosses this threshold. This indicates the percentage storage capacity used up and defaults to a value of 90, For example, the system sends an SNMP trap when the file system allocated space crosses 90% of the total storage capacity.

7.2 Network subsystem: *pvsVideoNetwork*

The network subsystem enumerates the status and configuration of all the network interfaces installed on the disk recorder for video transfers. The term network interface in this context necessarily refers to a physical network adapter and does not apply to instances like a software loop-back. It segregates the managed objects into a table defining the properties of all the interfaces and a group defining the managed objects for the video transfer protocols bound to these interfaces. The protocol group enumerates only the IP-stack binding configuration.

Table 11 – Video network interfaces table

Object	Syntax	Usage
PvsVnIfSlot	Integer32	The slot location of this network interface. The interface is uniquely identified using this slot number.
PvsVnIfDescr	DisplayString	A textual description of the network interface. This could indicate the manufacturer name, the model, and the version of the hardware interface.
PvsVnIfPhysAddr	PhysAddress	The data-link layer address for this interface or the interface's address immediately 'below' the network-layer in the protocol stack. For interfaces which do not have such an address (for example, a serial interface), this object should contain an octet string of zero length.
pvsVnIfSpeed	Gauge32	An estimate of the interface's current bandwidth in megabits per second. For interfaces that do not vary in bandwidth or for those where no accurate estimate can be made, this object should contain the nominal bandwidth.
pvsVnIfMtu	Integer32	The size of the largest data-gram which can be sent or received on the interface, specified in octets. For interfaces that are used for transmitting network data-grams, this is the size of the largest network data-gram that can be sent on the interface.
pvsVnIfType	INTEGER { other (1), fibreChannel (2), fastEther (3) }	The type of interface. Possible values for this variable are defined by the Internet Assigned Numbers Authority (IANA), through updating the syntax of the IANAifType textual convention. Refer the IANAifType-MIB for details.
pvsVnIfOutFrames	Counter32	The total number of frames transmitted out of the interface, including framing characters. NOTE: A frame in this context refers to the unit of transfer at the media-access layer. For ATM, this would refer to an ATM cell, and so on.
pvsVnIfOutErrors	Counter32	The number of outbound frames that could not be transmitted because of errors.
pvsVnIfInFrames	Counter32	The total number of frame received on the interface, including framing characters. NOTE: A frame in this context refers to the unit of transfer at the media-access layer. For ATM, this would refer to an ATM cell, and so on.
pvsVnIfInErrors	Counter32	The number of inbound frames that contained errors preventing them from being deliverable to a higher-layer protocol.
pvsVnIfState	INTEGER { unknown (1), up (2), down (3) }	The generic operational state of this video network interface.

The IP video network protocol group defines a table, every row of which is a binding configuration of the IP stack configuration with a particular interface. Thus, each entry is this table is uniquely identified by the slot location of the interface to which this IP configuration is bound.

Table 12 – IP configuration table

Object	Syntax	Usage
pvsVnIpIfBinding	Integer32	The slot number of the interface with which this protocol configuration is bound.
pvsVnIpAddress	IpAddress	The assigned IP host address.
pvsVnIpSubNetMask	IpAddress	The assigned IP subnet mask.
pvsVnIpDefaultGateway	IpAddress	The assigned default IP gateway address.

7.3 Video subsystem: *pvsVideo*

The video subsystem defines managed objects for all video signal-processing entities installed on the disk recorder. It enumerates the attributes of all the SDI input video channels (or physical BNC connectors) installed on the disk recorder. The slot location of the SDI I/O board and the channel location from top to bottom identify each row in this table.

Table 13 – SDI inputs table

Object	Syntax	Usage
pvsSdiSlot	Integer32	The PCI slot number of the input channel. This is a unique number from 1 to pvsCcNumSlots that identifies the SDI I/O board on which the input channel is present.
pvsSdiChannel	Integer32	The physical input channel identification number. The channel number on the top end of the board has the smallest numerical value and the value increases with each channel down the board. This uniquely identifies a channel on the SDI I/O board installed on slot pvsSdiSlot.
pvsSdiVideoDetected	INTEGER { video-absent (1), video-present (2) }	Indicates whether an external video signal is detected on the SDI input channel.
pvsSdiVitcDetected	INTEGER { vitc-absent (1), vitc-present (2) }	Indicates whether vertical interval time-code (VITC) was detected in the external video signal on this input channel.
pvsSdiAutoTimed	INTEGER { not-auto-timed (1), auto-timed (2) }	Indicates whether the input channel is configured for the auto-timed mode. When the input is in the auto-timed mode, the video signal is timed using the internal frame sync pulse. In the not-auto-timed mode the input allows the signal to pass asynchronously.
pvsSdiAudioDetectedG1	INTEGER { audio-absent (1), audio-present (2) }	Indicates whether embedded audio is detected in the first audio group on the SDI input channel.
pvsSdiAudioDetectedG2	INTEGER { audio-absent (1), audio-present (2)	Indicates whether embedded audio is detected in the second audio group on the SDI input channel.

	}	
pvsSdiAudioDetectedG3	INTEGER { audio-absent (1), audio-present (2) }	Indicates whether embedded audio is detected in the third audio group on the SDI input channel.
pvsSdiAudioDetectedG4	INTEGER { audio-absent (1), audio-present (2) }	Indicates whether embedded audio is detected in the fourth audio group on the SDI input channel.
pvsSdiAlias	DisplayString	The alias or name assigned by the user to the input channel.

7.4 Audio subsystem: *pvsAudio*

The audio group defines objects for all the audio processing and I/O entities within the disk recorder. This group defines a table of all the external audio options connected to the disk recorder. The disk recorder connects to the external audio option via its audio processing boards. The table has entries for all audio-processing boards installed in the system. Each row in the audio option table is thus uniquely identified by the slot location of the audio processing board.

Table 14 – Audio option table

Object	Syntax	Usage
pvsAuInterfaceSlot	Integer32	This PCI slot number of the audio board. This is a unique number from 4 to pvsCcNumSlots that identifies the audio interface to which the option is connected.
PvsAuOptionConnected	INTEGER { none (1), pac216 (2), pac208 (3), xlr216-or-bnc216 (4) }	Indicates whether an external option is connected to the audio an interface. If there is no external option attached, or there is no electrical connectivity between the external option and the audio interface, the value will be 'none(1)'. If a passive option (AES/EBU Breakout box) Is attached to the audio interface, the value will be 'xlr216-or-bnc216(4)'. An audio interface can recognize the presence of a PAC208 external option. However this option is not supported on the disk recorder.

7.5 Timing reference subsystem: *pvsTimingReference*

The objects in this group not only describe the particulars and configuration of the system's timing characteristics, but also include the status of the LTC input channels installed on the disk recorder.

Table 15 – Reference timing objects

Object	Syntax	Usage
pvsTrVideoMode	INTEGER { ntsc-525i-29-97Hz (1), pal-625i-25Hz (2), edtviii (3), unknown (4), hd-720p-59-94Hz (5) , hd-720p-60Hz (6) ,	Indicates the current video standard the system uses to play or record video. This object enumerates the conventional standard definition and all possible high definition standards that the disk recorder could support.

	hd-1080p-23-98Hz (7), hd-1080p-24Hz (8) , hd-1080p-25Hz (9), hd-1080p-29-97Hz (10) , hd-1080p-30Hz (11), hd-1080i-25Hz (12), hd-1080i-29-97Hz (13) , hd-1080i-30Hz (14), hd-1035i-30Hz (15) , hd-1035i-29-97Hz (16) }	
pvsTrVitcDetected	INTEGER { vitc-absent (1), vitc-present (2), }	Indicates whether VITC is present in the external reference timing signal.
pvsTrRefBlackDetected	INTEGER { reference-absent (1), reference-present (2), }	Indicates whether the disk recorder is locked to an external reference timing signal.
pvsTrRefSetting	INTEGER { other (1), bi-level-ref-black-29-97 (2), bi-level-ref-black-25 (3), tri-level-sync-1035i-30 (4), tri-level-sync-1035i-29-97 (5) , tri-level-sync-1080i-30 (6) , tri-level-sync-1080i-29-97 (7), tri-level-sync-1080i-25 (8) , tri-level-sync-1080p-30 (9), tri-level-sync-1080p-29-97 (10) , tri-level-sync-1080p-25 (11), tri-level-sync-1080p-24 (12), tri-level-sync-1080p-23-98 (13) , tri-level-sync-720p-60 (14), tri-level-sync-720p-59-94 (15) , hd-1035i-29-97Hz (16) }	Indicates the current external reference timing signal. The bi-level enumeration is for the standard definition video standards and the tri-level-sync enumeration is for the high definition video standards.

Table 16 – LTC input channel table

Object	Syntax	Usage
pvsTrLtcChannel	Integer32	The LTC channel or connector reference number as designated on the I/O panel.
pvsTrLtcDetected	INTEGER { ltc-absent (1), ltc-present (2) }	Indicates whether LTC was detected on the LTC channel.

7.6 Power supply subsystem: *pvsPowerSupply*

The disk recorder supports the redundant option of two system power supply units. The power supply group defines the power supply unit table that indicates the status of the upper and lower power supply slots on the disk recorder's chassis. Each power supply unit is identified by its slot location in the chassis. Each row in this table is indexed on the power supply slot location.

Table 17 – System power supply unit table

Object	Syntax	Usage
PvsPsUnit	INTEGER { upper-slot (1), lower-slot (2) }	A unique number that identifies a power supply unit provision in the system chassis.
PvsPsStatus	INTEGER { normal (1), faulty (2), not-installed (3) }	Indicates the operational condition of the power-supply. The faulty status could indicate conditions like an imminent power-supply shutdown, or that the unit is not receiving ac line voltage.

7.7 Thermal subsystem: *pvsThermal*

The thermal group defines the objects that monitor the thermal conditions of the system chassis. The group segregates the object definitions into object definitions for the system cooling fans and the internal chassis temperature specifics. The temperature group provides for setting the temperature thresholds at which the system sends warnings or alarms about abnormal thermal conditions.

Table 18 – Definition for the system cooling fan status

Object	Syntax	Usage
pvsFanStatus	INTEGER { fan-ok (1), fan-fault (2) }	Indicates the functional status of the fans on the system fan cage assembly. If all the fans are operating correctly, the status is 'fan-ok(1)'. The 'fan-fault(2)' state indicates that one or more fans are in a faulty condition, such as a stuck rotor or a missing fan assembly.

Table 19 – Thermal control and status objects

Object	Syntax	Usage
pvsTpWarningThreshold	Integer32	This specifies the temperature threshold value which the system uses for sending out overheating warnings. The threshold is specified as a temperature value in degrees Celsius. A change to this value takes effect only after a system reboot even though a subsequent SnmpGet operation returns the new value. The default threshold is 45 degrees Celsius.
pvsTpAlarmThreshold	Integer32	This specifies the temperature threshold value which the system uses for sending out overheated alarms. The threshold is specified as a temperature value in degrees Celsius. A change to this value takes effect only after a system reboot even though a subsequent SnmpGet operation returns the new value. The default threshold is 55 degrees Celsius.
pvsTpAmbientTemperature	Integer32	The current ambient temperature inside the card cage in degrees Celsius.

7.8 System disk subsystem: *pvsSystemDisk*

The disk recorder uses a mirrored disk drive system for redundant operation of its configuration and control operating system. The objects defined by this group describe the drive controller's operational status and the configuration of each of the mirrored system drives.

Table 20 – System drive controller status object

Object	Syntax	Usage
pvsSdControllerStatus	INTEGER { controller-not-detected (1), single-mode-operation (2), mirrored-mode-operation (3) }	Indicates the status of the system disk drive controller board that is used to mirror data being stored to the system drives. controller-not-detected(1) - The system is not able to detect the presence of the controller board. This may be due to bad cabling, an unresponsive adapter, or a board that is not installed. single-mode-operation(2) - The system has detected the presence of a controller board configured for the non-mirrored mode of operation. mirrored-mode-operation(3) - The system has detected the presence of a controller board configured for the mirrored mode of operation.

Table 21 – Mirrored system drive table

Object	Syntax	Usage
pvsSdDrive	INTEGER { primary-system-drive (1), mirrored-system-drive (2) }	A unique number that identifies a system disk drive unit. The primary drive is the drive that is inserted in the top system drive slot, and the mirrored drive is the one that is inserted in the bottom system drive slot.
pvsSdDriveConfig	INTEGER { active-drive (1), passive-drive (2) }	Indicates whether this drive is used for disk read operations when the system drives are operated in mirrored mode. Active-drive(1) - All disk reads are performed from this drive. Passive-drive(2) - This drive is used only for mirroring data during a disk write operation. No disk reads are performed from this drive when in the mirrored mode of operation. If the drive configured as the active disk fails, the drive controller automatically uses the passive drive for all disk read operations. By default, the primary system drive is set up as the active drive.
pvsSdDriveStatus	INTEGER { system-disk-ok (1), system-disk-bad (2) }	Indicates the operational status of a system disk drive.

8 DDRMON-MIB1 MIB notification definitions

The DDRMON-MIB1 MIB defines top-level notification definitions, each indicating a state change on the disk recorder. The DDRMON-MIB1 MIB also uses the *pvsLastTrapCategory* variable as the first variable in each of its trap definitions to indicate the basic severity level indicated by the notification. The following table lists the notification definitions.

Table 22 – DDRMON-MIB1 MIB notification definitions

Notification	Included Objects	Usage
pvsModeChange	• pvsLastTrapCategory • pvsMode	This trap indicates whether the system's operational mode has changed from the conventional mode to the maintenance mode, or vice versa. The maintenance mode typically indicates conditions such as diagnostics being run on the system, or some other user-initiated action that has impaired the normal behavior of the system. Similarly, when the system returns to its conventional mode, this trap is triggered to indicate that the system is back in normal operation. The maintenance mode is identified when the value of *pvsLastTrapCategory* equals 'warning(2)'. To indicate normal operation, the value of *pvsLastTrapCategory* equals 'informational(3)'.
pvsBoardStatusChange	• pvsLastTrapCategory • pvsCcBoardSlot • pvsCcBoardType • pvsCcBoardStatus • pvsCcBoardErrorMsg	This trap serves the generic purpose of indicating that something is wrong with a particular board in the card cage, or that a board with a previously-reported problem has resumed normal operation. The trap parameters indicate the type, slot number of the faulty board, and the error code that was sent by the board.
pvsTempStatusChange	• pvsLastTrapCategory • pvsTpAmbientTemperature	This trap is triggered whenever the ambient temperature inside the card cage overshoots the temperature thresholds, or the temperature returns to the recommended equipment operating range. When the temperature exceeds the value of *pvsTpWarningThreshold* , this trap is sent with the value of *pvsLastTrapCategory* set to 'warning'. Similarly when it exceeds the value of *pvsTpAlarmThreshold* the trap is sent with the value of *pvsLastTrapCategory* set to 'alarm(1)'. If the ambient temperature inside the chassis recedes to a value less than *pvsTpWarningThreshold*, this trap is sent out with the value of *pvsLastTrapCategory* set to 'informational(3)'.
pvsFanStatusChange	• pvsLastTrapCategory • pvsFanStatus	This indicates a failure on the system cooling fan assembly, or the cooling fan assembly returning to normal conditions. A fault can indicate errors such as a fan not installed, or a locked fan rotor. This trap does not identify the faulty fan or fans.

pvsPowerSupplyStatusChange	• pvsLastTrapCategory • pvsPsUnit • pvsPsStatus	This trap is triggered when the system encounters a fault in any of the installed power supply units, or the power supply starts operating normally after a fault. When this trap reports a 'warning(2)', it indicates that the system is running on the redundant (backup) power supply unit. The variables included in the trap identify the faulty power supply slot location, as well as the power supply unit's state.
PvsDriveModuleStatusChange	• pvsLastTrapCategory • pvsControllerId • pvsChassisId • pvsDriveId • pvsDriveStatus • pvsDriveRebuildStatus	This trap is triggered when a change is encountered on one of the drive modules in the system's RAID system. The parameters included in this trap indicate the faulty drive, it's location in the RAID chassis associated with a RAID controller, and the reason for the drive's failure. The system also uses this trap to indicate data rebuild progress on a particular RAID drive module.
PvsRaidChassisStatusChange	• pvsLastTrapCategory • pvsControllerId • pvsChassisId • pvsChassisStatus	This trap is triggered whenever an external RAID chassis reports a change in its operational state. The variables included in the trap identify the reporting chassis, along with its present state and operational severity level, as well as the controller that indicated the change in the chassis operating state.
PvsRaidControllerStatusChange	• pvsLastTrapCategory • pvsControllerId • pvsControllerStatus	This trap is triggered whenever the one of the RAID controllers on the external RAID chassis experiences a change in its operational state. The variables included with the trap indicate the severity level of the controller's present state along with its address and status.
PvsStorageCapacityStatusChange	• pvsLastTrapCategory • pvsFsBlockSize • pvsFsFreeBlocks • pvsFsUsedBlocks	This trap signals significant changes in the available media storage capacity. If the percentage of storage capacity remaining equals or is less than that indicated by *pvsFsCapacityThreshold*, this trap is triggered with the *pvsLastTrapCategory* set to value 'warning(2)'. If there is no storage space available and the system cannot record media, then this trap is triggered with *pvsLastTrapCategory* set to indicate an 'alarm(1)'. After any of these traps are sent, and media content on the storage system is lowered to a percentage lower than *pvsFsCapacityThreshold*, the trap will be triggered with *pvsLastTrapCategory* set to 'informational(3)' to indicate normal storage capacity.
PvsSystemDriveFailure	• pvsLastTrapCategory • pvsSdDrive	This trap is triggered when the system detects a failure on one of the system drives. The trap indicates a 'warning(2)' category and identifies the system drive that failed.

Annex A (informative)
MIB OID registration

A.1 OIDs for new designs

For new designs utilizing this MIB, the implementer should contact IANA (Internet Assigned Numbers Authority), which is the registration authority. They currently have a web site at: http://www.IANA.org

A.2 OIDs from the original design

The original objects defined in this MIB are located under the private enterprises subtree.

```
--  The  objects  defined  in  this   MIB   are   located   under   the
--  private.enterprises subtree as shown below:
--
--                    iso(1).org(3).dod(6).internet(1)
--                                 |
--                              private(4)
--                                 |
--                            enterprises(1)
--                                 |
--                              gvg(4947)
--                                 |
--                          gvgRegistrations(2)
--                                 |
--                           gvgVideoStorage(1)
--                                 |
--                              pvsMIB(2)
--                                 |
--                    +------------+------------+---------// ...etc...
--                    |                         |
--                    |                  +------------------+
--                pvsGeneral(1)          | pvsSubSystems(3) |
--                    |                  +------------------+
--                    |                           |
--                  ..etc..                    ..etc..
--
--  References
--
--
--  [1]  RFC 1907.  McCloghrie, K., and M. Rose, "Management Information
--       Base for Version 2 of the Simple Network Management Protocol
--       (SNMPv2)". January, 1996.
--
--  [2]  RFC 1902.  Rose, M., and K. McCloghrie, "Structure of Management
--       Information for Version 2 of the Simple Network Management
--       Protocol (SNMPv2)".  January, 1996.
--
--  [3]  Profile XP Media Platform System Guide.
--       Grass Valley Group Inc.
--
--  [4]  Profile XP Media Platform Service Manual.
--       Grass Valley Group Inc.
```

Annex B (informative)
Bibliography

ANSI/SMPTE 259M-1997, Television — 10-Bit 4:2:2 Component and 4fsc Composite Digital Signals — Serial Digital Interface

ANSI/SMPTE 299M-1997, Television — 24-Bit Digital Audio Format for HDTV Bit-Serial Interface

SMPTE 12M-1999, Television, Audio and Film — Time and Control Code

SMPTE 170M-1999, Television — Composite Analog Video Signal — NTSC for Studio Applications

SMPTE 272M-1994, Television — Formatting AES/EBU Audio and Auxiliary Data into Digital Video Ancillary Data Space

SMPTE 274M-2003, Television — 1920 x 1080 Image Sample Structure. Digital Representation and Digital Timing Reference Sequences for Multiple Picture Rates

SMPTE 292M-1998, Television — Bit-Serial Digital Interface for High-Definition Television Systems

SMPTE 296M-2001, Television — 1280 x 720 Progressive Image Sample Structure — Analog and Digital Representation and Analog Interface

SMPTE 305.2M-2000, Television — Serial Data Transport Interface (SDTI)

SMPTE 314M-1999, Television — Data Structure for DV-Based Audio, Data and Compressed Video - 25 and 50 Mb/s

SMPTE 318M-1999, Television — Synchronization of 59.94- or 50-Hz Related Video and Audio Systems in Analog and Digital Areas — Reference Signals

SMPTE 339M-2000, Television — Format for Non-PCM Audio and Data in AES3 — Generic Data Types

SMPTE 340M-2000, Television — Format for Non-PCM Audio and Data in AES3 — ATSC A/52 (AC-3) Data Type

AES3-1992 (R1997), Digital Audio Engineering — Serial Transmission Format for Two-Channel Linearly Represented Digital Audio Data

ANSI X3.230:1994, Information Technology — Fibre Channel — Physical and Signaling Interface (FC-PH)

ANSI X3.272:1996, Information Technology — Fibre Channel — Arbitrated Loop (FC-AL)

ANSI X3.287:1996, Information Technology — Fibre Channel — Link Encapsulation (FC-LE)

ANSI X3.297:1997, Information Technology — Fibre Channel — Physical and Signaling Interface — 2 (FC-PH-2)

ANSI/TIA/EIA-422-B Electrical Characteristics of Balanced Voltage Digital Interface Circuits (ANSI/TIA/EIA-422-B-94) (R2000)

ISO/IEC 8802-3:2000 (E), Information Technology - LAN/MAN - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications

ISO/IEC 13818-2:2000, Information Technology — Generic Coding of Moving Pictures and Associated Audio Information: Video

ITU-R (CCIR) Report 624-4:1990, Characteristics of Television Systems

GVG-ELEMENT-MIB, Grass Valley Group Inc., “Common Object Definitions for a Grass Valley Element”, December 1999

GVG-PVS-MIB, Grass Valley Group Inc., “Grass Valley Group Profile Video File Server MIB”, December 1999

GVG-REG, Grass Valley Group Inc., “Grass Valley Group Inc. Enterprise OID Registrations”, November 1999