RDD 4-2005

SMPTE REGISTERED DISCLOSURE DOCUMENT

Pro-AV MIB-1 – A Professional Audio/Visual Product Simple Network Management Protocol Management Information Base

This 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:

Toshiaki Setogawa
Sony Corporation
4-14-1 Asahicho, Atsugi
Kanagawa, 243-0014
Japan

E-mail Toshiaki.Setogawa@jp.sony.com

Approved
June 14, 2005

1 Scope

This document defines a Management Information Base (MIB) for Professional AV products as Pro-AV MIB-1 based upon Sony implementations. The MIB defines the items managed by the Simple Network Management Protocol (SNMP) and comprises variables that maintain the status of equipment.

Provisions of the Pro-AV MIB-1 provides capability for any company to create a private node, contents can be individually managed by that company.

2 Normative references

RFC 1155, Structure and Identification of Management Information for TCP/IP-based Internets

RFC 1157, A Simple Network Management Protocol

RFC 1213, Management Information Base for Network Management of TCP/IP-based internets: MIB-II

RFC 1903, Textual Conventions for Version 2 of the Simple Network Management Protocol

RFC 1907, Management Information Base for Version 2 of the Simple Network Management Protocol

RFC 2578, Structure of Management Information Version 2

RFC 2579, Textual Conventions for SMIv2

ITU-T Recommendation X.731, State Management Function

ITU-T Recommendation X.733, Alarm Reporting Function

3 Definitions

3.1 Agent: A software module that performs the network management functions requested by network management stations. An agent module may be implemented in any network element that is to be managed, such as a host, bridge, or router. Agents and network management stations communicate by means of SNMP.

3.2 Management Information Base (MIB): In the context of SNMP, this term is used in two contexts:

(1) A structured set of data variables, called objects, in which each variable represents some resource to be managed. Each agent in a network maintains a MIB for the network element on which it executes.

(2) The definition of a related collection of objects that represent some related collection of resources to be managed.

3.3 Object: A data variable that represents some resource or other aspect of a managed device; also referred to as a managed object.

3.4 Object Identifier (OID): Uniquely identifies an object within a MIB. The form of an object identifier is a sequence of numbers separated by periods. This sequence defines the location of an object in the tree-structured MIB of which it is a part.

3.5 Trap: An unsolicited message sent by an agent to a management station. The purpose is to notify the management station of some unusual event.

4 Pro-AV MIB-1 Overview

Pro-AV MIB-1 has a structure that can support monitoring of various products. Also, items and structures are simplified as much as possible in order to be able to easily implement for various categories of products.

The Structure of Management Information (SMI) for the MIB definitions shall comply with RFC 2578 (SMIv2).

The items composing the MIB should be grouped so that they can be managed in three sets of information as follows:

- Common information
- Category common information
- Individual product information

This document defines the common information.

NOTE – This MIB can be applied under each individual company ID, after setting an appropriate OID number for Pro-AV MIB-1 in parallel to existing OID's under the company OID.

5 Pro-AV MIB-1 Structure

The composition of Pro-AV MIB-1 structure shall be as follows.

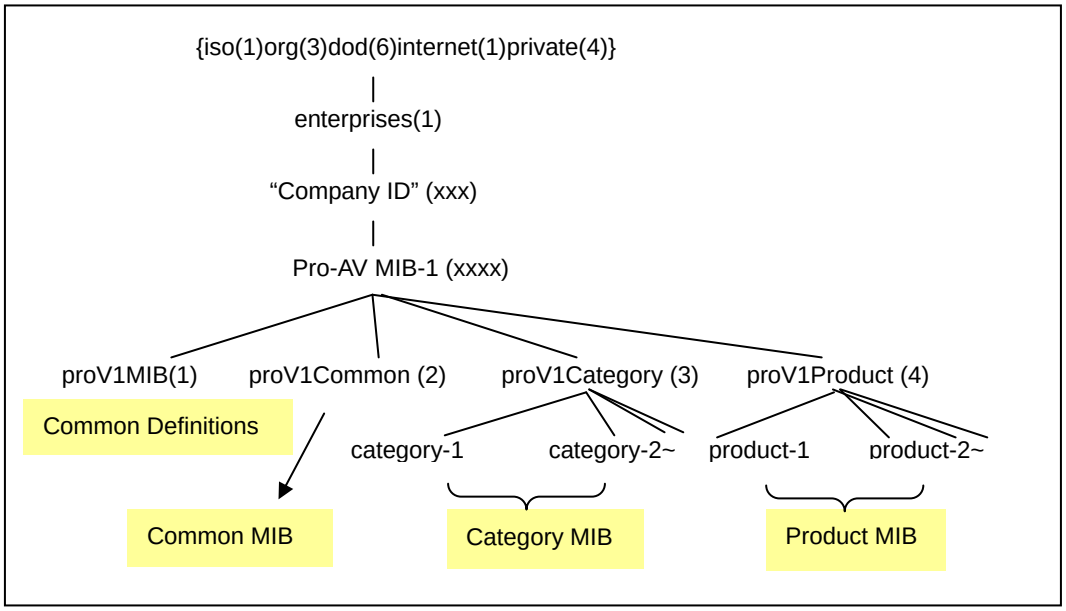

Figure 1 – Pro-AV MIB-1 Overall Structure

5.1 Common Definitions (proV1MIB)

Definitions commonly used in "Common MIB", "Category MIB" and "Product MIB" shall be entered. Consequently, the "TEXTUAL-CONVENTION" macro defining data expression is entered. This is not implemented as objects, but provided separately as a MIB definition file. An OID shall be given to the module as a Module Identifier by the rule of SMI.

5.2 Common MIB (proV1Common)

The variable items mounted in all products are defined in the Common MIB. The product ID information (including module information), information concerning the SNMP agent, error table, and trap definitions shall be included in the Common MIB.

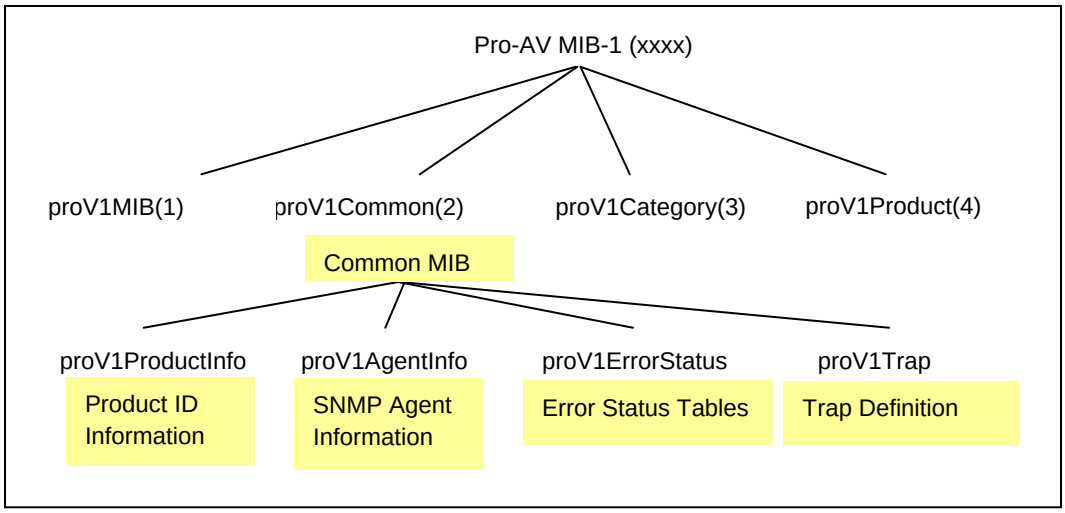

Figure 2 – Common MIB (proV1Common) Structure

5.3 Category MIB (proV1Category)

The common variable items for each category are defined in the Category MIB. When a new Category MIB is created, it shall be added positioned below this proV1Category. This shall be individually defined and managed by each company.

5.4 Product MIB (proV1Product)

The variable items for each product are defined in the Product MIB. Product MIBs shall be prepared when defining variable items unique to a product that are not defined by the Common MIB or Category MIB. When a new Product MIB is created, it is added below this proV1Product. This shall be individually defined and managed by each company.

6 Individual items in Common Definitions

The Common Definitions contain definitions of information common to the "Common MIB", "Category MIB", and "Product MIB". Consequently, the "TEXTUAL-CONVENTION" macro defining data expression is entered.

The TEXTUAL CONVENTION is a macro used to clearly express data for which the meaning may be different even though the data type is the same. For example, when there are frequencies and cycle times as numerical information, both may be expressed as integers, but the values shall have different meanings for each type. The following TEXTUAL CONVENTION shall be defined for such purposes as clarifying these differences.

(1) Local time

This expresses the local time information as a character string with the format "YYYY/MM/DD hh:mm:ss:ff GMT+HH:MM". The ff positioned after the seconds (ss) expresses the frame number, and

the section from GMT onward expresses the time difference from Greenwich Mean Time (GMT). The + may be replaced by –.

```
ProV1TimeStamp ::= TEXTUAL-CONVENTION
                    STATUS current
                    DESCRIPTION
                          "textual information for printable timestamp(local time).
                          example: 2001/08/29 22:30:00:00 GMT+09:00"
                    SYNTAX DisplayString (SIZE(32))
```

(2) Cycle time
This expresses a cycle time as an integer in the range 0-2147483647.

```
ProV1Cycle ::= TEXTUAL-CONVENTION
                STATUS current
                DESCRIPTION
                    "textual information for cycle time
                     unit: sec."
                SYNTAX Integer32 (0..2147483647)
```

(3) Frequency
This expresses a frequency as an integer in the range 0-2147483647.

```
ProV1Frequency ::= TEXTUAL-CONVENTION
                    STATUS current
                    DESCRIPTION
                          "textual information for frequency
                           unit: /sec."
                    SYNTAX Integer32 (0..2147483647)
```

(4) Scanning line number
This expresses the scanning line number as an integer in the range 0-2147483647.

```
ProV1ScanningLineNumber ::= TEXTUAL-CONVENTION
                    STATUS current
                    DESCRIPTION
                          "textual information for scanning line number
                           unit: /sec"
                    SYNTAX Integer32 (0..2147483647)
```

(5) Frame number
This expresses the frame number as an integer in the range 0-2147483647.

```
ProV1FrameNumber ::= TEXTUAL-CONVENTION
                    STATUS current
                    DESCRIPTION
                        "textual information for frame number
                         unit: /sec"
                    SYNTAX Integer32 (0..2147483647)
```

(6) Device status
This expresses the device ON/OFF status as unknown (1), down (2) or up (3).

```
ProV1Status ::= TEXTUAL-CONVENTION
                STATUS current
                DESCRIPTION
                    "textual information for device status."
                SYNTAX INTEGER { unknown(1), down(2), up(3) }
```

The first character of the name of the TEXTUAL CONVENTION shall always start with the capital letter P. (ProV1Abcdef).

7 Individual items in Common MIB

The Common MIB shall include the following items and item tables.

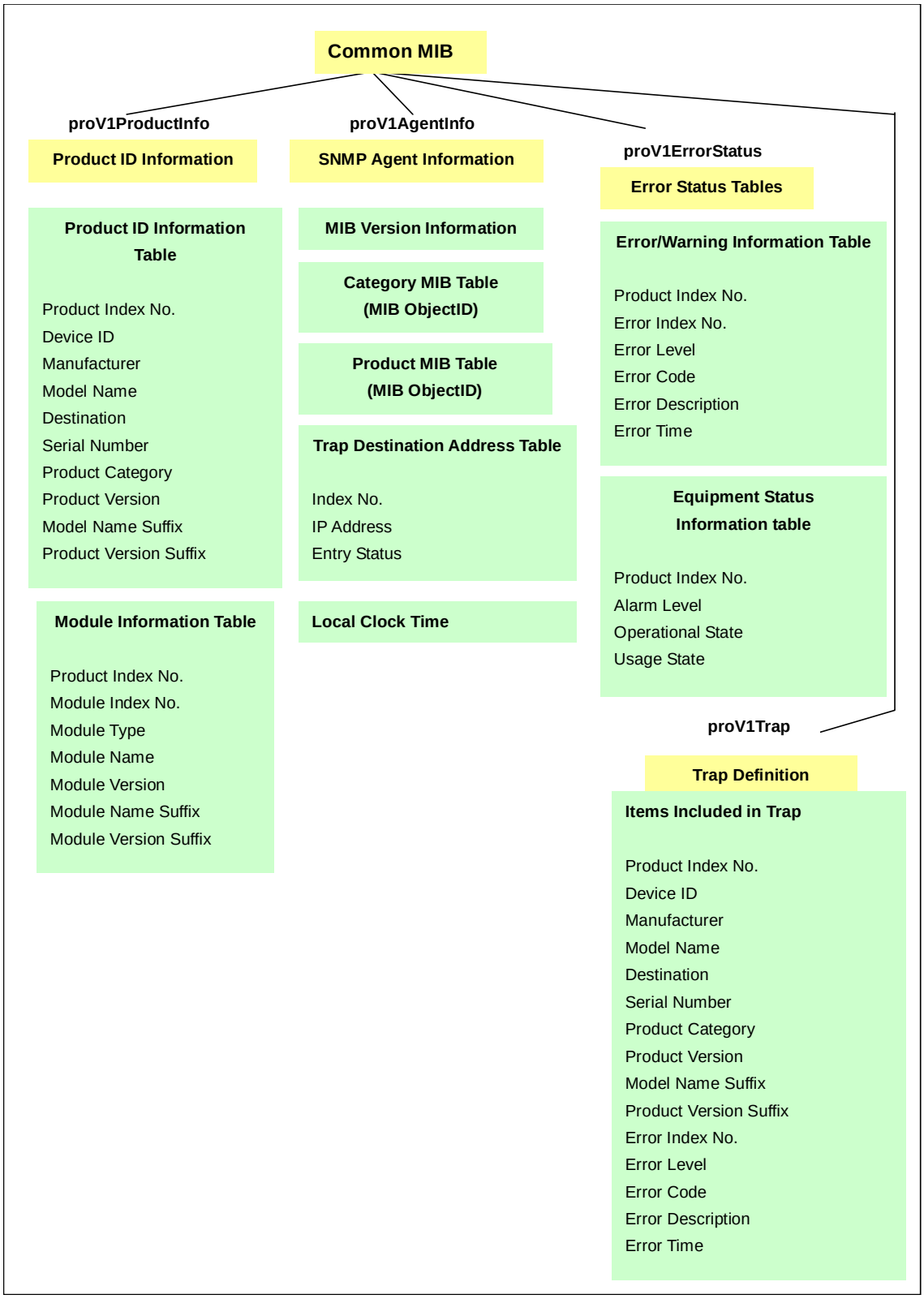

Figure 3 – Items included in Common MIB

Items are read-only status unless otherwise stated.

Also, items that take DisplayString (alphanumeric) as the variable type shall never include the following five characters: <, >, &, ", '.

Table 1 – Definition of OID

Parents	No.	MIB Name	R/W	Update Timing
enterprises	(Note*6) XXX	XXXX (Note*7)	-	----(OID) (Note*5)
XXXX (Note*7)	xxxx (Note*8)	Pro-AV MIB-1 (proAVMIB1)	-	----(OID) (Note*5)
(Following are the 1st layer)				
Pro-AV MIB-1 (proAVMIB1)	1	proV1MIB	-	----(OID) (Note*5)
	2	proV1Common	-	----(OID) (Note*5)
	3	proV1Category	-	----(OID) (Note*5)
	4	proV1Product	-	----(OID) (Note*5)
	5	proV1Trap	-	----(OID) (Note*5)
(Following are detailed Information for common portion)				
proV1Common	1	proV1ProductInfo	-	----(OID) (Note*5)
	2	proV1AgentInfo	-	----(OID) (Note*5)
	3	proV1ErrorStatus	-	----(OID) (Note*5)
(Following are Product ID Information)				
proV1ProductInfo	1	proV1ProdIdInfoTable	-	----
	2	proV1ModuleTable	-	----
proV1ProdIdInfoTable	1	proV1ProdIdInfoEntry	-	----
proV1ProdIdInfoEntry	1	proV1ProdIdInfoIdx	RO	(Note *2)
	2	proV1ProdIdInfoDevId	RO	(Note *3)
	3	proV1ProdIdInfoManufact	RO	(Note *2)
	4	proV1ProdIdInfoModel	RO	(Note *2)
	5	proV1ProdIdInfoDest	RO	(Note *2)
	6	proV1ProdIdInfoSN	RO	(Note *2)
	7	proV1ProdIdInfoCat	RO	(Note *2)
	8	proV1ProdIdInfoVer	RO	(Note *2)
	9	proV1ProdIdInfoModelSuffix	RO	(Note *2)
	10	proV1ProdIdInfoVerSuffix	RO	(Note *2)
proV1ModuleTable	1	proV1ModuleEntry	-	----
proV1ModuleEntry	1	proV1ModuleProdId	RO	(Note *2)
	2	proV1ModuleIdx	RO	(Note *2)
	3	proV1ModuleType	RO	(Note *2)
	4	proV1ModuleName	RO	(Note *2)
	5	ProV1ModuleVer	RO	(Note *2)
	6	proV1ModuleNameSuffix	RO	(Note *2)
	7	proV1ModuleVerSuffix	RO	(Note *2)

(continued)

(Following are SNMP Agent Information)				
proV1AgentInfo	1	proV1AgentMIBVer	RO	(Note *1)
	2	proV1AgentMIBCatTable	-	----
	3	proV1AgentMIBProdTable	-	----
	4	proV1TrapDestTable	-	----
	5	(Reserved) (Note *9)	-	----
	6	proV1LocalClockTime	RO	(Note *4)
proV1AgentMIBCatTable	1	proV1AgentMIBCatEntry	-	----
proV1AgentMIBCatEntry	1	proV1AgentMIBCatIdx	RO	(Note *1)
	2	proV1AgentMIBCatOID	RO	(Note *1)
proV1AgentMIBProdTable	1	proV1AgentMIBProdEntry	-	----
proV1AgentMIBProdEntry	1	proV1AgentMIBProdIdx	RO	(Note *1)
	2	proV1AgentMIBProdOID	RO	(Note *1)
proV1TrapDestTable	1	proV1TrapDestEntry	-	----
proV1TrapDestEntry	1	proV1TrapDestIdx	RO	(Note *4)
	2	proV1TrapDestAddress	R/C	(Note *4)
	3	proV1TrapDestEntryStatus	R/C	(Note *4)
(Following are Error Status)				
ProV1ErrorStatus	1	ProV1ErrStatTable	RO	(Note *1)
	2	proV1ErrSttEqSttTable	RO	(Note *1)
proV1ErrStatTable	1	proV1ErrStatEntry	-	---
ProV1ErrStatEntry	1	proV1ErrStatProdId	RO	(Note *4)
	2	proV1ErrStatErrIdx	RO	(Note *4)
	3	proV1ErrStatLevel	RO	(Note *4)
	4	proV1ErrStatCode	RO	(Note *4)
	5	proV1ErrStatDescr	RO	(Note *4)
	6	proV1ErrStatTime	RO	(Note *4)
ProV1ErrSttEqSttTable	1	proV1ErrSttEqSttEntry	-	
ProV1ErrSttEqSttEntry	1	proV1ErrSttEqSttProdId	RO	(Note *4)
	2	ProV1ErrSttEqSttAlmLevel	RO	(Note *4)
	3	proV1ErrSttEqSttOprStt	RO	(Note *4)
	4	proV1ErrSttUsgStt	RO	(Note *4)
(Following are Trap Definition)				
proV1Trap	65793	proV1Trap1	-	---
proV1Trap1		proV1ProdIdInfoIdx		
		proV1ProdIdInfoDevId		
		proV1ProdIdInfoManufact		
		proV1ProdIdInfoModel		
		proV1ProdIdInfoDest		
		proV1ProdIdInfoSN		
		proV1ProdIdInfoCat		
		proV1ProdIdInfoVer		
		proV1ProdIdInfoModelSuffix		
		proV1ProdIdInfoVerSuffix		
		proV1ErrStatErrIdx		
		proV1ErrStatLevel		
		proV1ErrStatCode		
		proV1ErrStatDescr		
		proV1ErrStatTime		

Comments:

About R/W (read/write attribute):

R/C:	Read/Create
R/W:	Read/Write
RO:	Read Only
-:	Not Accessible

About Update timing:

Note *1)

Shipping from factory: A fixed value is set when the equipment is shipped from the factory, improved, or upgraded.

Note *2)

Shipping/configuration change: The value is reset when the equipment is shipped from the factory as above or for a proxy product when the configuration of the equipment underneath it is changed.

Note *3)

User setting: The value is set by the user that using the control panel or http.

Note *4)

Dynamic: The equipment program periodically monitors the status and sets the value.

About Descriptions

Note *5)

----(OID): Object identifier; does not take a value.

Note *6)

XXX: The Company ID.
e.g. 122 for Sony Corporation.

Note *7)

XXXX: The Company Name.
e.g. sony

Note *8)

XXXX: The number assigned within the company.
e.g. 8501 has been assigned for Sony's Pro-AV MIB-1.

Note *9)

(Reserved) Can be defined freely by each company.

7.1 Product ID Information

The Product ID Information comprises the ID information table for specifying the product and the module information table for acquiring the information for the modules included in the product.

7.1.1 Product ID Information Table

Stores the Product ID Information in table format.

proV1ProdIdInfoTable
proV1ProdIdInfoEntry

(1)	Index No.	proV1ProdIdInfoIdx	Integer32
(2)	Device ID	proV1ProdIdInfoDevId	DisplayString
(3)	Manufacturer	proV1ProdIdInfoManufact	DisplayString
(4)	Model Name	proV1ProdIdInfoModel	DisplayString
(5)	Destination	proV1ProdIdInfoDest	DisplayString
(6)	Serial Number	proV1ProdIdInfoSN	DisplayString
(7)	Product Category	proV1ProdIdInfoCat	INTEGER
(8)	Product Version	proV1ProdIdInfoVer	Integer32
(9)	Model Name Suffix	proV1ProdIdInfoModelSuffix	DisplayString
(10)	Product Version Suffix	proV1ProdIdInfoVerSuffix	DisplayString

Stand-alone Product:

If the product is a stand-alone product that is connected as a stand-alone unit to the network, this table has only one entry.

Table 2 – Example of Stand-alone Product Connected to Network as Stand-alone Unit

Index No.	Device ID	Manufacturer	Model Name	Destination	Serial Number	Product Category	Product Version
1	11	XXXX	DXX-500	XX	100102	1100	110

Proxy Product/System Product:

When the product has stand-alone products underneath it, and send/receive information to/from the network in place of the stand-alone products, the product also has entries of stand-alone products such as the following.

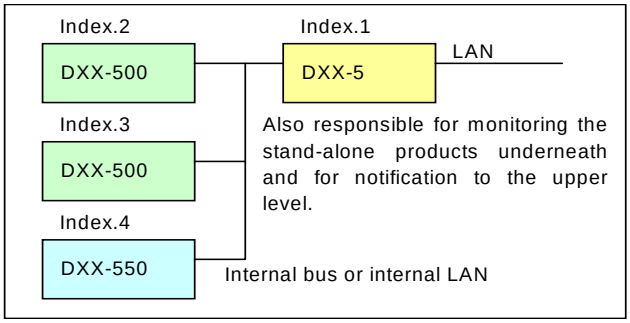

Figure 4 – Proxy product/system product

Table 3 – Example in which the product has stand-alone products underneath it

Index No.	Device ID	Manufacturer	Model Name	Destination	Serial Number	Product Category	Product Version
1	01	XXXX	DXX-5	XX	110012	9000	200
2	11	XXXX	DXX-500	XX	100102	1100	110
3	12A	XXXX	DXX-500	XX	100103	1100	110
4	13A	XXXX	DXX-550	XX	100109	1100	130

(1) Index No.
This is an index number for specifying the entries in this table.
(proV1ProdIdInfoIdx)(variable type = Integer32)

(2) Device ID
A device number or name is entered that is set so that the customer can identify the product.
It is assumed that the device ID is either set numerically from the control panel of the product or that a character string is set using http or telnet through the network.
Also, it is mounted so that the Device ID in the first line (instance) of this table takes the same value as sysName in the MIB-II System Group.
(proV1ProdIdInfoDevId) (variable type = DisplayString, maximum 255 characters)

(3) Manufacturer
The vendor name is entered.
(proV1ProdIdInfoManufact) (variable type = DisplayString, maximum 64 characters)

(4) Model Name
The model name is entered.
Also, it is mounted so that the Model Name in the first line (instance) of this table takes the same value as sysDescr in the MIB-II System group.
(proV1ProdIdInfoModel) (variable type = DisplayString, maximum 255 characters)

(5) Destination
The destination is entered.
(proV1ProdIdInfoDest) (variable type = DisplayString, maximum 64 characters)

(6) Serial Number
The serial number is entered.
(proV1ProdIdInfoSN) (variable type = DisplayString, maximum 64 characters)

(7) Product Category
Category type groups are defined in table 4. Only the major categorization is defined. The detail categorization shall be specified and managed by each company.
(proV1ProdIdInfoCat) (variable type = INTEGER)

Table 4 – Product Category

No.	Category Type Group	Example of Product Categories
1000~1999	Storage Products	VTR, DISK RECORDER, DATA RECORDER
2000~2999	Processing Products	ROUTER, SWITCHER, EFFECTS, PERIPHERAL, CONTROLLER, CODEC, EDITOR
3000~3999	Camera Products	STUDIO CAMERA, SECURITY CAMERA, CAMCORDER
4000~4999	Display Products	MONITOR, FLAT PANEL MONITOR, PROJECTOR, LARGE DISPLAY
5000~5999	Audio Products	AUDIO MIXER, CD, DAT, MIC
6000~6999	AV Other Products	PRINTER
7000~7999	Test and Measurement	WFM, SG, MPEG TEST SYSTEM
8000~8999	PC and Network Products	PC BASED PRODUCT, PC SERVER, VIDEO CONFERENCE
9000~9999	Other Products	

(8) Product Version
The version number for the product is entered. This is the version set for the product as a whole, not the version for the internal firmware or the like.
(proV1ProdIdInfoVer) (variable type = Integer32)

(9) Model Name Suffix
Extension of the model name is entered, if necessary.
(proV1ProdIdInfoModelSuffix) (variable type = DisplayString, maximum 64 characters)

(10) Product Version Suffix
Extension of the version number for the product is entered, if necessary.
(proV1ProdIdInfoVerSuffix) (variable type = DisplayString, maximum 64 characters)

NOTE – For a description of the Product Version and Module Version, a combination of two kinds of expressions are provided. One is the Integer32 (version) to make it easy to track the order. The other is the DisplayString (Version Suffix) to allow free spelling. This combined system can be operated by each company policy.

7.1.2 Module Information Table

The module information included in the product shall be stored in a module information table in table form. This table is primarily used to store the versions for the firmware (ROM) included in the product. This information can also be used to read out information on the hardware configuration of the product.

proV1ModuleTable
proV1ModuleEntry

(1) Product ID index	proV1ModuleProdId	Integer32
(2) Index No.	proV1ModuleIdx	Integer32
(3) Module Type	proV1ModuleType	INTEGER
(4) Module Name	proV1ModuleName	DisplayString
(5) Module Version	proV1ModuleVer	Integer32
(6) Module Name Suffix	proV1ModuleNameSuffix	DisplayString
(7) Module Version Suffix	proV1ModuleVerSuffix	DisplayString

The combination of (1) and (2) serves as the index for this table.

(1) comes before (2) in order to raise the search performance of the SNMP Manager.

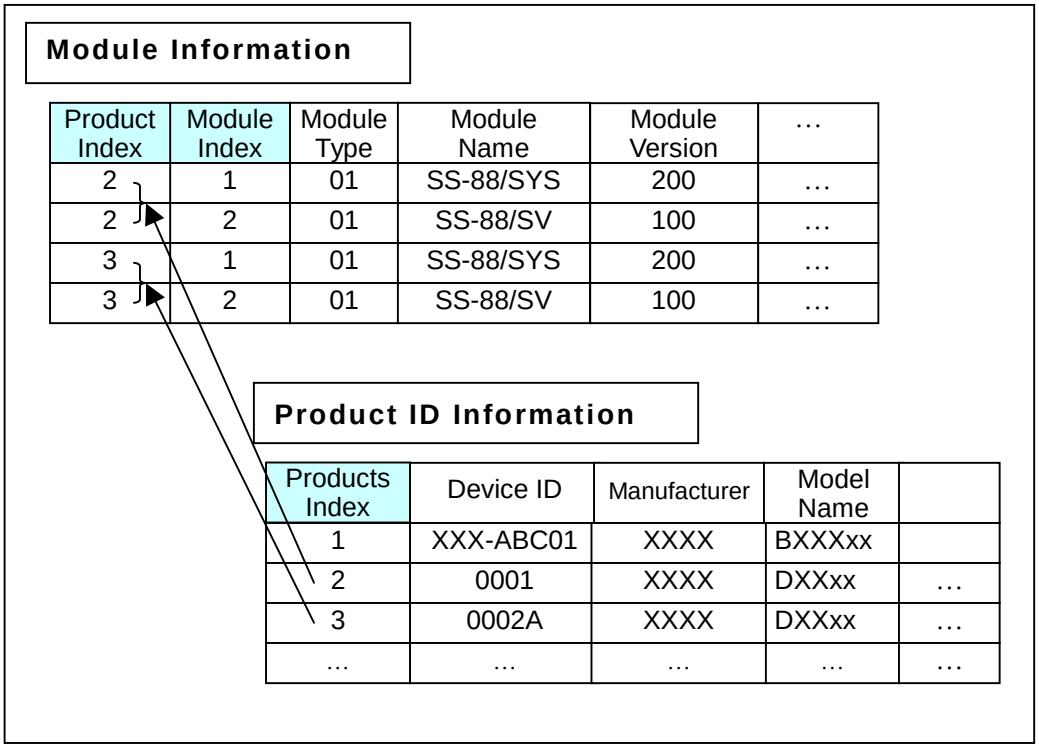

Figure 5 – Relationship between Module Information Table and Product ID Information Table

(1) Product ID Index (Product Index)
When the information of the multiple box products is included in the Product ID Information Table described above, this Product ID Index shows which box product each module belongs to. When the table is only for one product, 1 is entered.
(proV1ModuleProdId) (variable type = Integer32)

(2) Index No. (Module Index)
This is an index number for specifying the entry in this table. This index number does not necessarily need to be consecutive. For example, ROM management number is determined according to the mounting location can be mounted, and if there is an empty mounting location, the corresponding index number can be skipped.
(proV1ModuleIdx) (variable type = Integer32)

(3) Module Type
The module type is entered. Codes should be defined and managed by each company.
(proV1ModuleType) (variable type = INTEGER)

(4) Module Name
The module name is entered.
For firmware, the board name/firmware name (ROM name) is appropriate.
(proV1ModuleName)(variable type = DisplayString, maximum 64 characters)

(5) Module Version
The module version number is entered.
(proV1ModuleVer)(variable type = Integer32)

(6) Module Name Suffix
Extension of the module name is entered, if necessary.
(ProV1ModuleNameSuffix)(variable type = DisplayString, maximum 64 characters)

(7) Module Version Suffix
Extension of the version number for the module is entered, if necessary.
(ProV1ModuleVerSuffix)(variable type = DisplayString, maximum 64 characters)

NOTE – Items that take DisplayString (alphanumeric) as the variable type shall never include the following five characters: <, >, &, ", '.

7.2 SNMP Agent Information

This is the MIB for information related to the SNMP agent. It comprises the MIB version information, Category MIB, and Product MIB mounting information, trap destination address table and local clock time. This information is used to make it easy to set and operate the SNMP Manager.

7.2.1 MIB Version Information

This MIB shall be written with the version Structure of Management Information (SMI) that is displayed. 100 shall be applied for SMIv1, and 200 for SMIv2.
(proV1AgentMIBVer) (variable type = Integer32)

7.2.2 Mounted Category MIB Table

This table shows which expansion MIB are mounted in the SNMP agent in addition to the Common MIB. This table contains information on the expanded MIB for each category. The top object identifier of the mounted Category MIB is held in this table. In the case of complex products, it may have plural Category MIB modules.

proV1AgentMIBCatTable
proV1AgentMIBCatEntry

(1) Index No.	proV1AgentMIBCatIdx	Integer32
(2) MIB OID	proV1AgentMIBCatOID	OBJECT IDENTIFIER

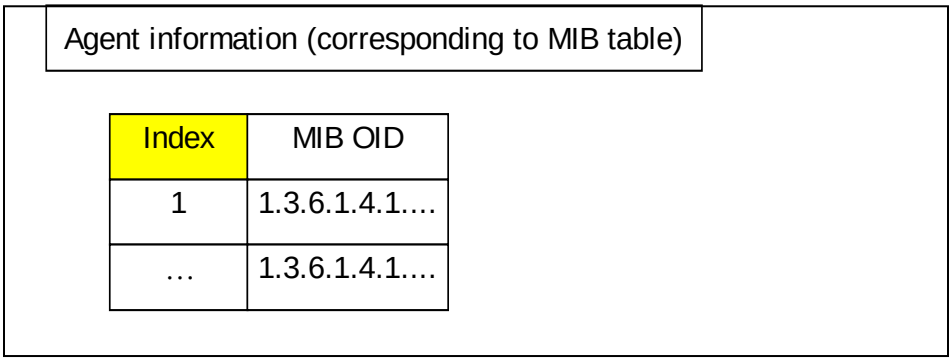

Figure 6 – Example of MIB Table

(1) Index No.
This is an index number for specifying the entry in this table.
(proV1AgentMIBCatIdx) (variable type = Integer32)

(2) MIB OID
This is the value of the object identifier for the MIB module that the equipment (agent) holds.
(proV1AgentMIBCatOID) (variable type = OBJECT IDENTIFIER)

7.2.3 Mounted Product MIB Table

This table shows which expanded MIB is mounted in the SNMP agent in addition to the Common MIB. This table contains information on expanded MIB for each product. The top object identifier of the mounted Product MIB module is held in this table. In the case of complex product, it may have multiple Product MIB modules. The structure of this table is the same as the structure of the Category MIB table.

proV1AgentMIBProdTable		
proV1AgentMIBProdEntry		
(1) Index No.	proV1AgentMIBProdIdx	Integer32
(2) MIB OID	proV1AgentMIBProdOID	OBJECT IDENTIFIER

(1) Index No.
This is an index number for specifying the entry in this table.
(proV1AgentMIBProdIdx) (variable type = Integer32)

(2) MIB OID
This is the value of the object identifier for the MIB module that the equipment (agent) holds.
(proV1AgentMIBProdOID) (variable type = OBJECT IDENTIFIER)

7.2.4 Trap Destination Address Table

The trap destination address (IP address) for SNMP agent is stored. The trap destination address is generally set up manually using the control panel, web browser, telnet, or the like when the product is installed. For some products, it is desirable to automate this process and simplify the installing operation. For such products, the structure takes into account setting up the trap destination address automatically through negotiation with the SNMP Manager.

proV1TrapDestTable		
proV1TrapDestEntry		
(1) Index No.	proV1TrapDestIdx	Integer32
(2) Trap Destination Address	proV1TrapDestAddress	IpAddress(Read-Create)
(3) Entry status	proV1TrapDestEntryStatus	INTEGER (Read-Create)

For some equipment, in which the automatic setting of the trap destination address is not desired, the SNMP agent may just mount the display of the trap destination address set in the equipment.

(1) Index No.
This is an index number for specifying the entry in this table.
(proV1TrapDestIdx) (variable type = Integer32)

(2) Trap Destination Address
The trap destination address is displayed.
(proV1TrapDestAddress) (variable type = IpAddress) (* Read-Create attribute)

(3) Entry status
This variable is for adding entries from the SNMP Manager. The status 5 (fixed) has been added to the definition of the EntryStatus generally used in network system industry for managing table format of the MIB.
(proV1TrapDestEntryStatus) (variable type = INTEGER) (* Read-Create attribute)

Table 5 – Definition of Entry Status Value

Value	Meaning
1 (valid)	Valid
2 (create Request)	Request to add row entry (from Manager to Agent)
3 (under Creation)	Indicates end of row entry creation operation (from Agent to Manager)
4 (invalid)	Invalid (This status is not generated.)
5 (fixed)	Shows that a fixed IP address is set.

7.2.5 Local Clock Time

When the unit has a built-in clock, its time is returned. In case the product does not have year, month, day or time information, “9999” or “99” should be inserted in each column.
(proV1LocalClockTime) (variable type = ProV1TimeStamp)

7.3 Error Status Tables

7.3.1 Error/Warning Information Table

Information on the errors and warnings currently generated in the product shall be stored in table format. This is not a table storing a history of past error/warning information. When the error/warning is ended, its information is deleted from the table. At that time, information of the end of error/warning may be temporarily displayed on the table.

proV1ErrStatTable
proV1ErrStatEntry

(1) Error Product Index	proV1ErrStatProdId	Integer32
(2) Index No.	proV1ErrStatErrIdx	Integer32
(3) Error Level	proV1ErrStatLevel	INTEGER
(4) Error Code	proV1ErrStatCode	DisplayString
(5) Error Description	proV1ErrStatDescr	DisplayString
(6) Error Time	proV1ErrStatTime	ProV1TimeStamp

The combination of (1) and (2) serves as the index for this table.

The (1) comes before (2) in order to raise the search performance of the SNMP Manager.

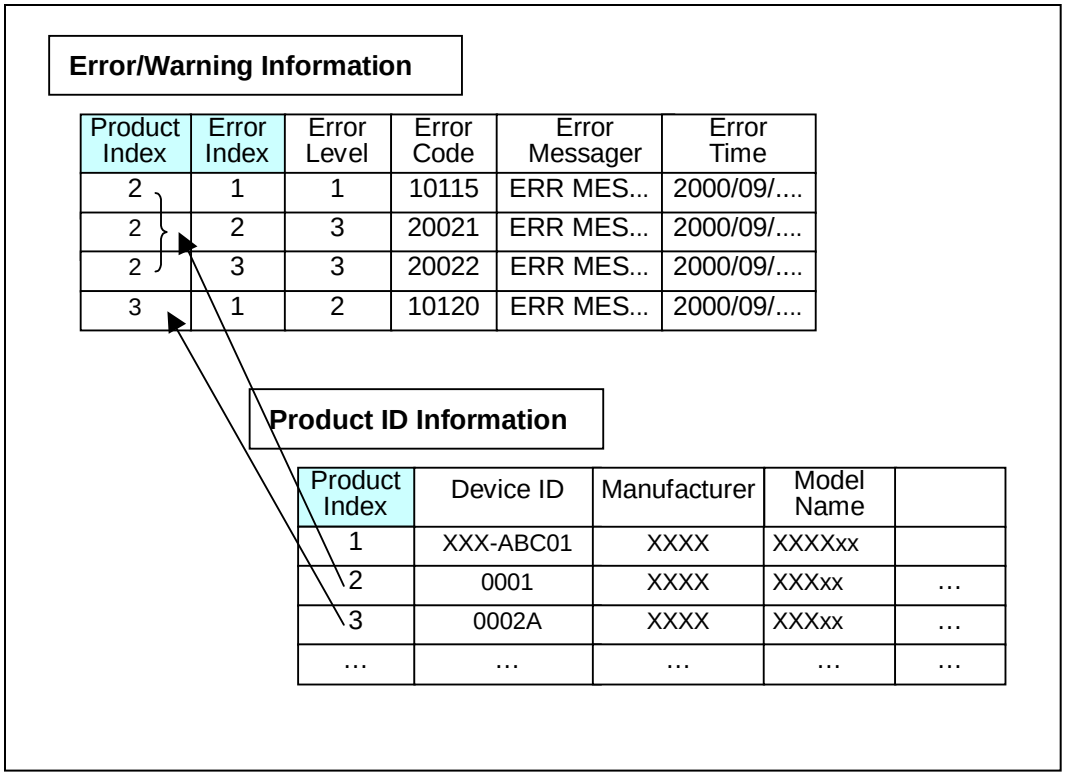

Figure 7 – Relationship between Error/Warning Information Table and Product ID Information Table

(1) Error Product Index
The index number of the Product ID Information Table (described above) shall be entered to specify the product that the error/warning is generated in.
If the product is a stand-alone product and has no other product underneath it, 1 is entered as the index value.
(proV1ErrStatProdId)(variable type = Integer32)

(2) Index No.
When managing faults within the product, an index number to specify the fault shall be entered.
For example, when three faults (described below) are detected, the corresponding error index numbers of (i), (ii), or (iii) are attached in order to manage and transmit the fault information.

(i) Interface port connection error
(ii) Power supply fan rotation rate drop
(iii) Picture compression process crash

As time passes, if fault (ii) is changed to "power supply fan stop", the error index No. = 2 is attached and this fault announced. The receiving side of this information (monitoring software) can change (ii) "Power supply fan rotation rate drop" to (ii) "power supply fan stop" in the fault management database that it holds, then the same fault can be tracked and managed.
(proV1ErrStatErrIdx) (variable type = Integer32)

(3) Error Level
A value shall be entered showing how critical the error is. Meaning of each severity level conforms to ITU-T X.733.

1 Critical:
It is necessary to stop the unit quickly and solve the error.

2 Major:
The error fault is not fatal, but is important.

3 Minor:
A state occurred that was not fatal but for which a warning should be issued.

4 Warning:
A state occurred for which notification should be issued for maintenance or operation.

5 Clear (Normal):
Error/warning ended
This is used when it is intended to display recovery from error/warning with the timing information on the table or when Trap is transmitted to notify the end of error/warning. Refer to 7.4 for Trap.

6 Information:
When the equipment (Agent) intends to notify some information which is not categorized to any error or warning, set the error level to this value and display it on the table and/or transmit the trap. Refer to 7.4 for Trap.
(proV1ErrStatLevel) (variable type = INTEGER)

(4) Error Code
An error code shall be entered that is prescribed for the product and product category.

Alphanumeric characters can be used, but capital letters and small letters are not distinguished. Also, do not use symbols.

Also, because the codes take the form of character strings not numbers, the same numeric value written in a different way is recognized as a different code, for example 0001 and 1.
(proV1ErrStatCode) (variable type = DisplayString, maximum 64 characters)

(5) Error Description
An explanation of the error or error name shall be entered with which the gist of the meaning of the error/warning can be understood.
(proV1ErrStatDescr) (variable type = DisplayString, maximum 255 characters)

(6) Error Time
The time at which the error/warning (or the end of error/warning) was generated shall be entered (yyyy/mm/dd hh:mm:ss:ff GMT+xx:xx). Here, the ff expresses the frame number and the GMT+xx:xx expresses the time difference from Greenwich Mean Time (GMT). In case of the product which does not have year, month, day or time information, “9999” or “99” should be filled in each column.
(proV1ErrStatTime) (variable type = ProV1TimeStamp (fixed at 32 bytes))

NOTE – Items that take DisplayString (alphanumeric) as the variable type shall not include the following five characters: <, >, &, ", '.

7.3.2 Equipment Status Information Table

Information on the equipment alarms currently generated in the product is stored in table format. This is not a table storing a history of past equipment alarm information.

proV1ErrSttEqSttTable
proV1ErrSttEqSttEntry

(1) Equipment Status Product Index	proV1ErrSttEqSttProdId	Integer32
(2) Alarm Level	proV1ErrSttEqSttAlmLevel	INTEGER
(3) Operational State	proV1ErrSttEqSttOprStt	INTEGER
(4) Usage State	proV1ErrsttEqSttUsgStt	INTEGER

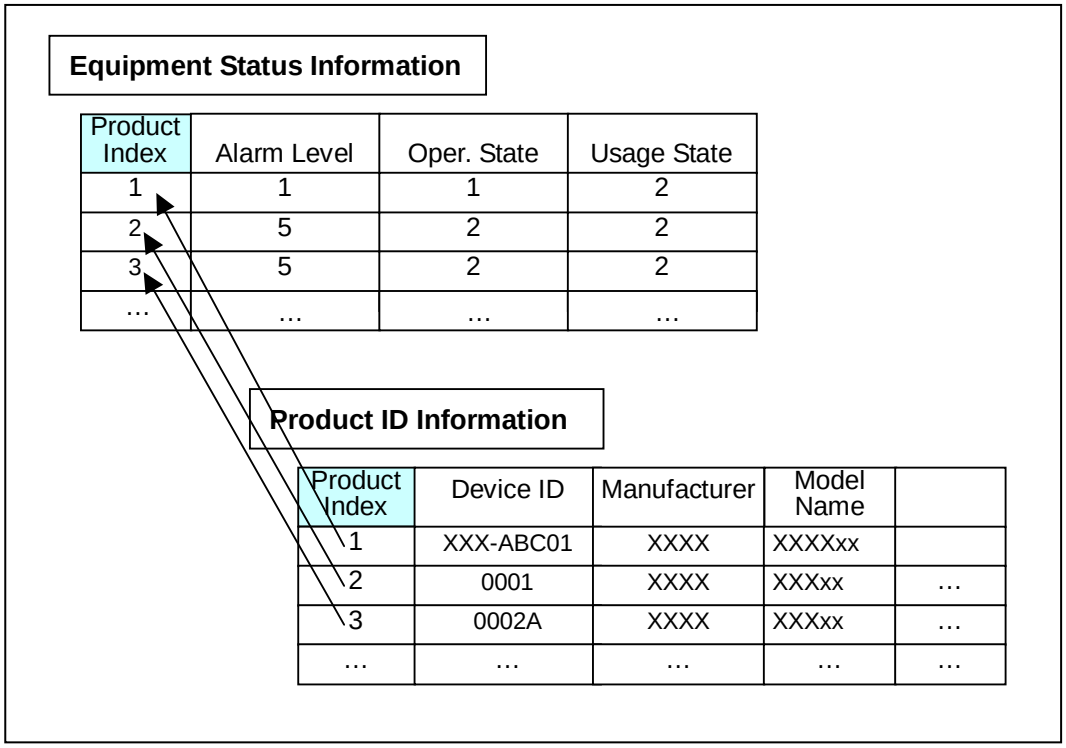

Figure 8 – Relation between Equipment Status Information Table and Product ID Information Table

(1) Equipment Status Product Index
The index number of the Product ID Information Table (described above) shall be entered to specify the product that the equipment status is generated in. If the product is a stand-alone product and has no other product underneath it, 1 is entered as the index value.
(proV1ErrSttEqSttProdId)(variable type = Integer32)

(2) Alarm Level
A value shall be entered showing how critical the alarm is. Meaning of each severity level conforms to ITU-T X.733.

1 Critical:
It is necessary to stop the unit quickly and solve the error.

2 Major:
The fault is not fatal, but is important.

3 Minor:
A state occurred that was not fatal but for which an alarm should be issued.

4 Warning:
A state occurred for which notification should be issued for maintenance or operation.

5 Normal (Cleared):
Normal State.

(proV1ErrSttEqSttAlmLevel) (variable type = INTEGER)

(3) Operational State
The operational state of the equipment shall be entered. The values are classifited into 2 states as follows. Meaning of each state conforms to ITU-T X.731.
1 Disabled
2 Enabled
(proV1ErrSttEqSttOprStt) (variable type = INTEGER)

(4) Usage State
The usage state of the equipment shall be entered. The values are classified into 3 states as follows. The meaning of each state conforms to ITU-T X.731.
1 Idle
2 Active
3 Busy
(proV1ErrSttEqSttUsgStt) (variable type = INTEGER)

7.4 Trap Definitions

7.4.1 Trap Definitions (proV1Trap1)

The trap announcing the occurrence (and end) of error/warning is defined as collections of the items below. The number “65793” is allocated for the Specific Trap of this trap, so the Agent shall be mounted so the trap is sent including this number in the PDU.

NOTE – This specific trap number = 65793 is determined with the method below.

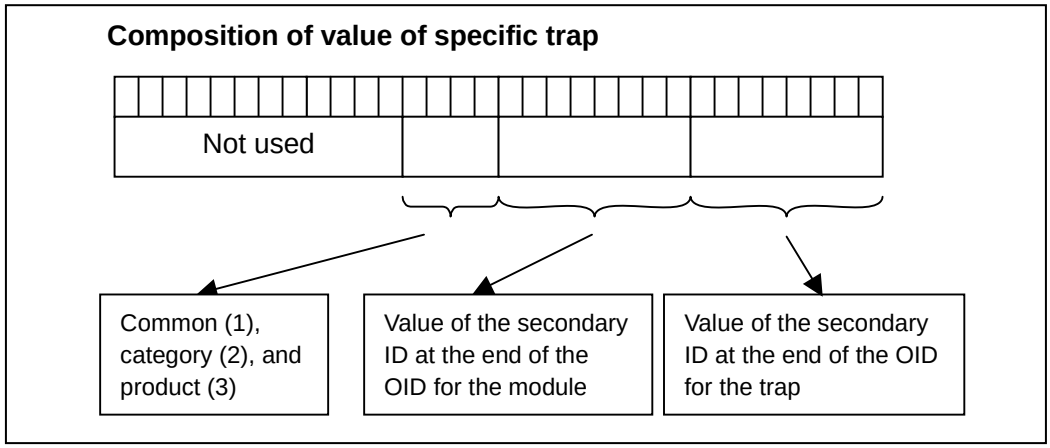

Figure 9 – Rules for Allocating Specific Trap Numbers for the Pro-AV MIB-1

In order to be the trap in the first line of the first MIB module defined in the Common MIB, 0001 0000 0001 0000 0001 (binary) = 65536 + 256 + 1 = 65793.

The variable (Variable Bindings) items included in the trap packet (PDU: Protocol Data Units) are included in the Product ID Information Table and Error Information Table described above, so for details refer to their individual explanations.

(1)	Product ID Index (Products Index)	proV1ProdIdInfoIdx	Integer32
(2)	Device ID	proV1ProdIdInfoDevID	DisplayString
(3)	Manufacturer	proV1ProdIdInfoManufact	DisplayString
(4)	Model Name	proV1ProdIdInfoModel	DisplayString
(5)	Destination	proV1ProdIdInfoDest	DisplayString
(6)	Serial Number	proV1ProdIdInfoSN	DisplayString
(7)	Product Category	proV1ProdIdInfoCat	INTEGER
(8)	Product Version	proV1ProdIdInfoVer	Integer32
(9)	Model Name Suffix	proV1ProdIdInfoModelSuffix	DisplayString
(10)	Product Version Suffix	proV1ProdIdInfoVerSuffix	DisplayString
(11)	Error Index Number	proV1ErrStatErrIdx	Integer32
(12)	Error Level	proV1ErrStatLevel	INTEGER
(13)	Error Code	proV1ErrStatCode	DisplayString
(14)	Error Description	proV1ErrStatDescr	DisplayString
(15)	Error Time	proV1ErrStatTime	ProV1TimeStamp

7.4.2 Trap PDU structure

Two kinds of the trap PDU (Protocol Data Units) structure are defined in figures 10 and 11. One conforms to the SNMPv1-Trap. The other conforms to the SNMPv2-Trap. Either can be adopted.

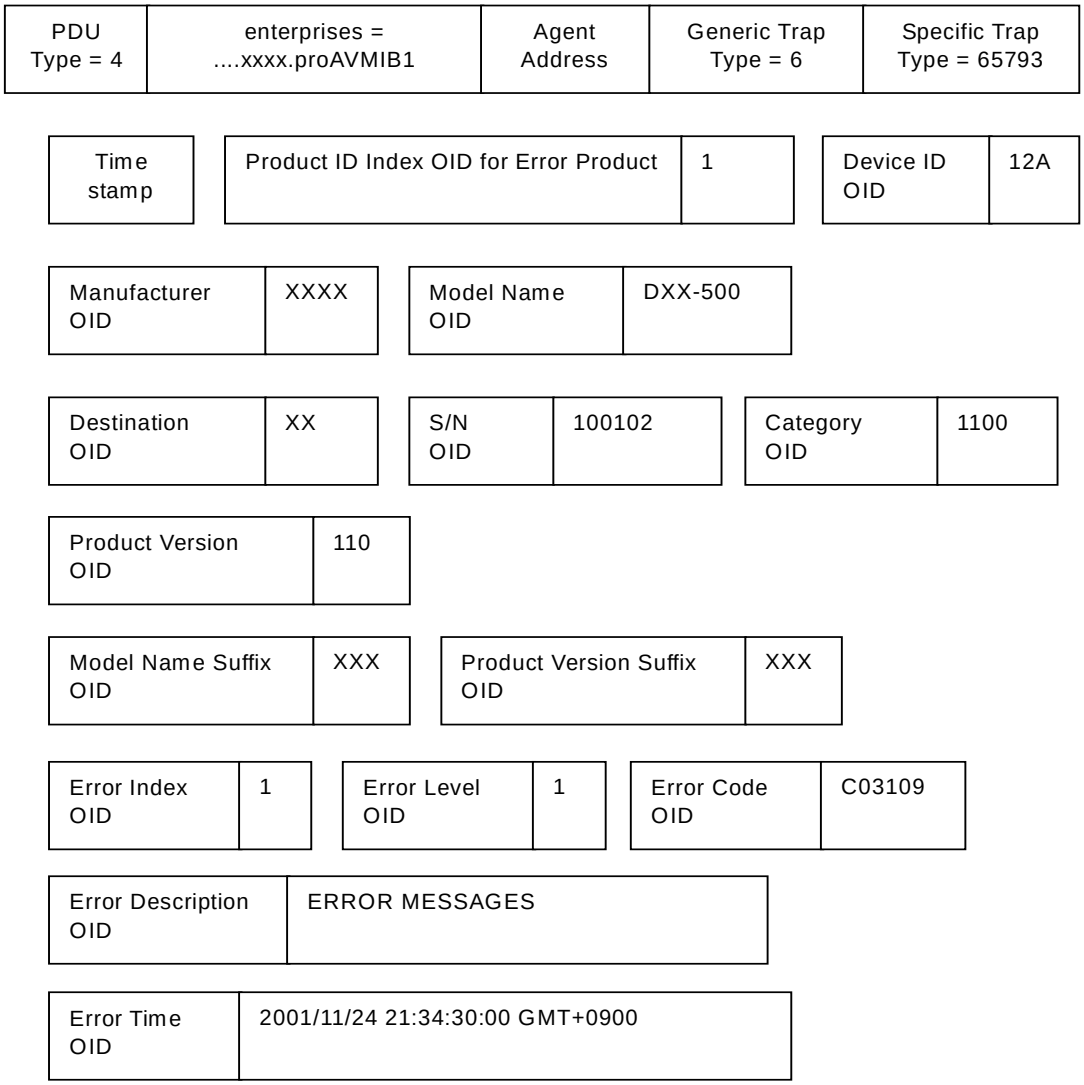

Figure 10 – SNMPv1-Trap-PDU Structure (the Variable Bindings values are reference examples)

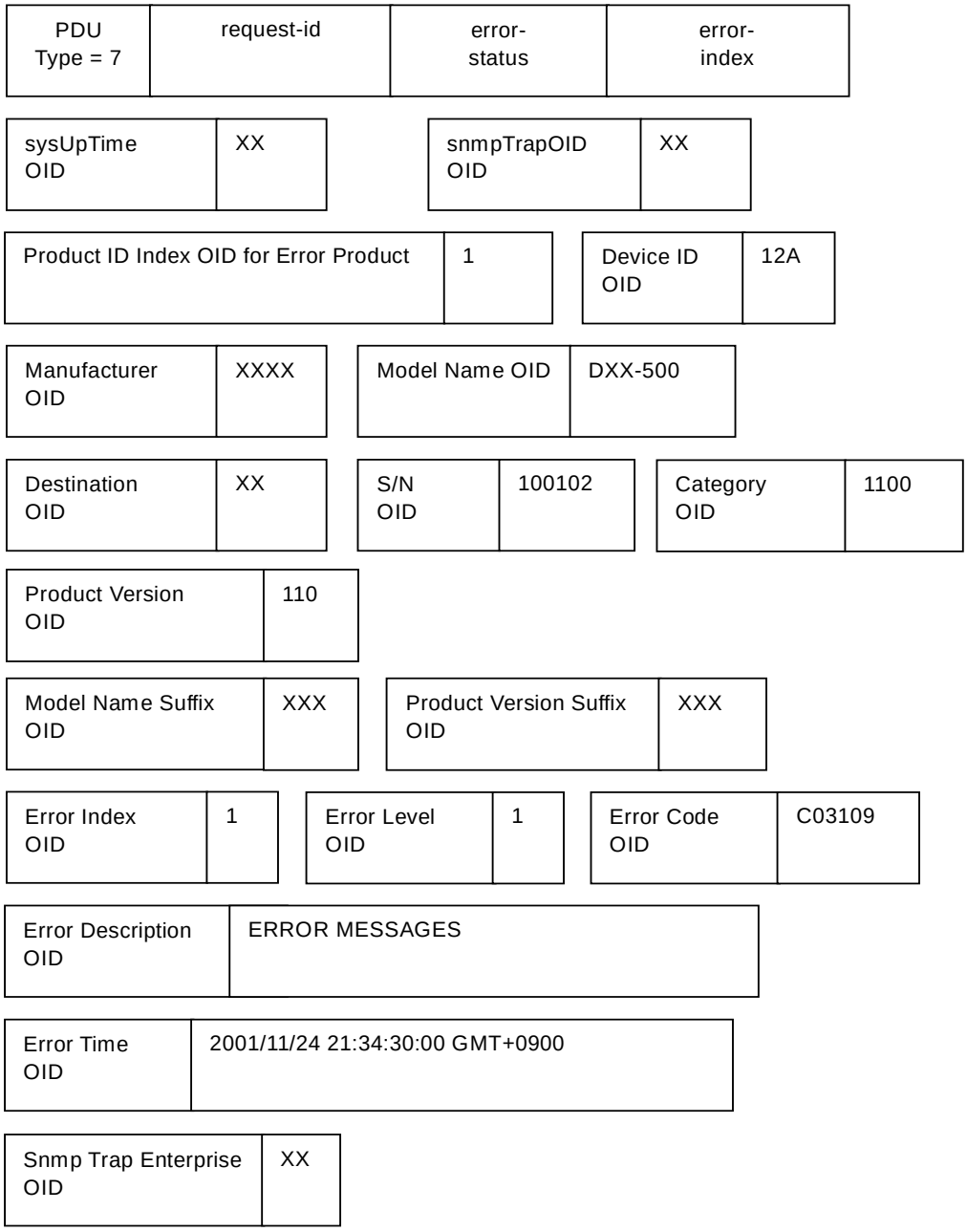

Figure 11 – SNMPv2-Trap-PDU Structure (the Variable Bindings values are reference examples.)

NOTE – sysUpTime, snmpTrapOID and snmpTrapEnterprise conform to MIB-II.

Annex A (informative)
Sample of Common Definitions

```
--
-- Pro-AV-MIB-1-Def-Sony
--
-- The Common Definitions for Pro-AV MIB-1.
--

Pro-AV-MIB-1-Def-Sony DEFINITIONS ::= BEGIN

IMPORTS
          TEXTUAL-CONVENTION
          DisplayString
                    FROM SNMPv2-TC
          enterprises,
          MODULE-IDENTITY,
          Integer32
                    FROM SNMPv2-SMI
          proAVMIB1
                    FROM SONY-SMI;

proV1MIB MODULE-IDENTITY
          LAST-UPDATED "200410231000Z"
          ORGANIZATION "Sony Corporation"
          CONTACT-INFO
                    "Professional Solutions Network Company,
                    Sony Corporation"
          DESCRIPTION
                    "The Common Definitions for Pro-AV MIB-1.
                    The Pro-AV MIB-1 is designed for use in managing
                    Professional Audio/Visual products."
          REVISION "200410231000Z"
          DESCRIPTION
                    "Initial revision"
          ::= { proAVMIB1 1 }


ProV1Cycle ::= TEXTUAL-CONVENTION
          STATUS current
          DESCRIPTION
                    "textual information for cycle time
                    unit: sec."
          SYNTAX Integer32 (0..2147483647)


ProV1FrameNumber ::= TEXTUAL-CONVENTION
          STATUS current
          DESCRIPTION
                    "textual information for the frame number
                    unit: /sec"
          SYNTAX Integer32 (0..2147483647)
```

```
ProV1Frequency ::= TEXTUAL-CONVENTION
          STATUS current
          DESCRIPTION
                    "textual information for frequency
                    unit: /sec."
          SYNTAX Integer32 (0..2147483647)


ProV1ScanningLineNumber ::= TEXTUAL-CONVENTION
          STATUS current
          DESCRIPTION
                    "textual information for the scanning line number
                    unit: /sec"
          SYNTAX Integer32 (0..2147483647)


ProV1Status ::= TEXTUAL-CONVENTION
          STATUS current
          DESCRIPTION
                    "textual information for device status"
          SYNTAX INTEGER {
                              unknown(1),
                              down(2),
                              up(3) }


ProV1TimeStamp ::= TEXTUAL-CONVENTION
          STATUS current
          DESCRIPTION
                    "textual information for printable timestamp(local
                    time).
                    example: 2001/08/29 22:30:00:00 GMT+09:00.
                    This expresses the local time information as a
                    character string with the format 'YYYY/MM/DD
                    hh:mm:ss:ff GMT+HH:MM'. The ff positioned after the
                    seconds (ss) expresses the frame number, and the
                    section from GMT onward expresses the time difference
                    from Greenwich Mean Time (GMT). The + may be replaced
                    by -."
          SYNTAX DisplayString (SIZE (32))


END
```

Annex B (Informative)
Sample of Common MIB

```
--
-- Pro-AV-MIB-1-Common-Sony
--
-- The Common MIB of Pro-AV MIB-1. The variable items mounted in all products
-- are defined in the Common MIB. The product ID information (including
-- module information), information concerning the SNMP agent, error table,
-- and trap definitions shall be included in the Common MIB.
--

Pro-AV-MIB-1-Common-Sony DEFINITIONS ::= BEGIN

IMPORTS
          DisplayString
                    FROM SNMPv2-TC
          OBJECT-GROUP,
          NOTIFICATION-GROUP,
          MODULE-COMPLIANCE
                    FROM SNMPv2-CONF
          ProV1TimeStamp
                    FROM Pro-AV-MIB-1-Def-Sony
          MODULE-IDENTITY,
          OBJECT-IDENTITY,
          OBJECT-TYPE,
          NOTIFICATION-TYPE,
          Integer32,
          IpAddress
                    FROM SNMPv2-SMI
          proAVMIB1
                    FROM SONY-SMI;

proV1Common MODULE-IDENTITY
          LAST-UPDATED "200410241329Z"
          ORGANIZATION "Sony Corporation"
          CONTACT-INFO
                    "Professional Solutions Network Company,
                    Sony Corporation"
          DESCRIPTION
                    "proV1Common is root OBJECT IDENTIFIER of
                    the Common MIB."
          REVISION "200410241329Z"
          DESCRIPTION
                    "Initial revision"
          ::= { proAVMIB1 2 }


proV1ProductInfo OBJECT-IDENTITY
          STATUS current
          DESCRIPTION
                    "The Product ID Information comprises the ID
                    information table for specifying the product
                    and the module information table for acquiring
                    the information for the modules included in
```

```
                    the product."
          ::= { proV1Common 1 }

proV1ProdIdInfoTable OBJECT-TYPE
          SYNTAX SEQUENCE OF ProV1ProdIdInfoEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
                    "Stores the Product ID Information in table
                    format."
          ::= { proV1ProductInfo 1 }


proV1ProdIdInfoEntry OBJECT-TYPE
          SYNTAX ProV1ProdIdInfoEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
                    "An entry in the proV1ProdIdInfoTable."
          INDEX {
                    proV1ProdIdInfoIdx }
          ::= { proV1ProdIdInfoTable 1 }


ProV1ProdIdInfoEntry ::= SEQUENCE {

          proV1ProdIdInfoIdx         Integer32,
          proV1ProdIdInfoDevID       DisplayString,
          proV1ProdIdInfoManufact    DisplayString,
          proV1ProdIdInfoModel       DisplayString,
          proV1ProdIdInfoDest        DisplayString,
          proV1ProdIdInfoSN          DisplayString,
          proV1ProdIdInfoCat         INTEGER,
          proV1ProdIdInfoVer         Integer32,
          proV1ProdIdInfoModelSuffix DisplayString,
          proV1ProdIdInfoVerSuffix   DisplayString }


proV1ProdIdInfoIdx OBJECT-TYPE
          SYNTAX Integer32 (1..2147483647)
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "This is an index number for specifying the
                    entries in this table."
          ::= { proV1ProdIdInfoEntry 1 }


proV1ProdIdInfoDevID OBJECT-TYPE
          SYNTAX DisplayString (SIZE (0..255))
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "A device number or name is entered that is
```

```
                    set so that the customer can identify the
                    product. It is assumed that the device ID is
                    either set numerically from the control panel
                    of the product or that a character string is
                    set using http or telnet through the network.
                    Also, it is mounted so that the Device ID in
                    the first line (instance) of this table takes
                    the same value as sysName in the MIB-2 System
                    Group."
          ::= { proV1ProdIdInfoEntry 2 }


proV1ProdIdInfoManufact OBJECT-TYPE
          SYNTAX DisplayString (SIZE (0..64))
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "The vendor name is entered."
          ::= { proV1ProdIdInfoEntry 3 }


proV1ProdIdInfoModel OBJECT-TYPE
          SYNTAX DisplayString (SIZE (0..255))
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "The model name is entered. Also, it is mounted
                    so that the Model Name in the first line
                    (instance) of this table takes the same value
                    as sysDescr in the MIB-2 System group."
          ::= { proV1ProdIdInfoEntry 4 }


proV1ProdIdInfoDest OBJECT-TYPE
          SYNTAX DisplayString (SIZE (0..64))
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "The destination is entered."
          ::= { proV1ProdIdInfoEntry 5 }


proV1ProdIdInfoSN OBJECT-TYPE
          SYNTAX DisplayString (SIZE (0..64))
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "The serial number is entered."
          ::= { proV1ProdIdInfoEntry 6 }


proV1ProdIdInfoCat OBJECT-TYPE
          SYNTAX INTEGER {
                              vtr(1100),
```

```
                                disk(1200),
                                dataRecorder(1300),
                                cart(1400),
                                discRecorder(1500),
                                router(2100),
                                swicther(2200),
                                dme(2300),
                                peripheral(2400),
                                controller(2500),
                                codec(2600),
                                editor(2700),
                                nonlinearEditor(2750),
                                systemCamera(3100),
                                securityCamera(3200),
                                ieee1394Camera(3220),
                                camcorder(3300),
                                monitor(4100),
                                flatpanelMonitor(4150),
                                projector(4200),
                                jjtron(4300),
                                informationDisplay(4400),
                                audioMixer(5100),
                                mo(5200),
                                cd(5220),
                                dat(5240),
                                mic(5300),
                                printer(6100),
                                pcBasedProduct(8100),
                                pcServer(8200),
                                videoConference(8300) }
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
                        "The category type information."
            ::= { proV1ProdIdInfoEntry 7 }


proV1ProdIdInfoVer OBJECT-TYPE
            SYNTAX Integer32
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
                        "The version number for the product is entered.
                        This is the version set for the product as a
                        whole, not the version for the internal firmware
                        or the like."
            ::= { proV1ProdIdInfoEntry 8 }


proV1ProdIdInfoModelSuffix OBJECT-TYPE
            SYNTAX DisplayString (SIZE (0..64))
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
```

```
                    "Extension of the model name is entered, if
                    necessary."
          ::= { proV1ProdIdInfoEntry 9 }


proV1ProdIdInfoVerSuffix OBJECT-TYPE
          SYNTAX DisplayString (SIZE (0..64))
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "Extension of the version number for the product
                    is entered, if necessary."
          ::= { proV1ProdIdInfoEntry 10 }


proV1ModuleTable OBJECT-TYPE
          SYNTAX SEQUENCE OF ProV1ModuleEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
                    "The module information included in the product
                    shall be stored in a module information table
                    in table form.  This table is primarily used
                    to store the versions for the firmware (ROM)
                    included in the product. This information can
                    also be used to read out information on the
                    hardware configuration of the product."
          ::= { proV1ProductInfo 2 }


proV1ModuleEntry OBJECT-TYPE
          SYNTAX ProV1ModuleEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
                    "An entry of proV1ModuleTable."
          INDEX {
                    proV1ModuleProdId,
                    proV1ModuleIdx }
          ::= { proV1ModuleTable 1 }


ProV1ModuleEntry ::= SEQUENCE {

          proV1ModuleProdId     Integer32,
          proV1ModuleIdx        Integer32,
          proV1ModuleType       INTEGER,
          proV1ModuleName       DisplayString,
          proV1ModuleVer        Integer32,
          proV1ModuleNameSuffix DisplayString,
          proV1ModuleVerSuffix  DisplayString }


proV1ModuleProdId OBJECT-TYPE
```

```
            SYNTAX Integer32 (1..2147483647)
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
                        "When the information of the multiple box
                        products is included in the Product ID
                        Information Table described above, this
                        Product ID Index shows which box product each
                        module belongs to. When the table is only for
                        one product, 1 is entered."
            ::= { proV1ModuleEntry 1 }


proV1ModuleIdx OBJECT-TYPE
            SYNTAX Integer32 (1..2147483647)
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
                        "This is an index number for specifying the
                        entry in this table.  This index number does
                        not necessarily need to be consecutive.  For
                        example, ROM management number is determined
                        according to the mounting location can be
                        mounted, and if there is an empty mounting
                        location, the corresponding index number can
                        be skipped."
            ::= { proV1ModuleEntry 2 }


proV1ModuleType OBJECT-TYPE
            SYNTAX INTEGER {
                                    rom(1),
                                    hardDisk(2),
                                    softwareModule(3),
                                    board(11),
                                    daughterBoard(12),
                                    functionModule(13) }
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
                        "The module type information."
            ::= { proV1ModuleEntry 3 }


proV1ModuleName OBJECT-TYPE
            SYNTAX DisplayString (SIZE (0..64))
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
                        "The module name is entered. For firmware, the
                        board name/firmware name (ROM name) is
                        appropriate."
            ::= { proV1ModuleEntry 4 }
```

```
proV1ModuleVer OBJECT-TYPE
          SYNTAX Integer32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "The module version number is entered."
          ::= { proV1ModuleEntry 5 }


proV1ModuleNameSuffix OBJECT-TYPE
          SYNTAX DisplayString (SIZE (0..64))
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "Extension of the module name is entered, if
                    necessary."
          ::= { proV1ModuleEntry 6 }


proV1ModuleVerSuffix OBJECT-TYPE
          SYNTAX DisplayString (SIZE (0..64))
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "Extension of the version number for the module
                    is entered, if necessary."
          ::= { proV1ModuleEntry 7 }


proV1AgentInfo OBJECT-IDENTITY
          STATUS current
          DESCRIPTION
                    "This is the MIB for information related to
                    the SNMP agent. It comprises the MIB version
                    information, Category MIB and Product MIB
                    mounting information, trap destination
                    address table and local clock time. This
                    information is used to make it easy to set and
                    operate the SNMP manager."
          ::= { proV1Common 2 }

proV1AgentMIBVer OBJECT-TYPE
          SYNTAX Integer32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "This MIB shall be written with the version
                    Structure of Management Information (SMI) that
                    is displayed. 100 shall be applied for SMIv1, and
                    200 for SMIv2."
          ::= { proV1AgentInfo 1 }
```

```
proV1AgentMIBCatTable OBJECT-TYPE
          SYNTAX SEQUENCE OF ProV1AgentMIBCatEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
                    "This table shows which expansion MIB are
                    mounted in the SNMP agent in addition to the
                    Common MIB.  This table contains information
                    on the expanded MIB for each category. The top
                    object identifier of the mounted Category MIB
                    is held in this table. In the case of complex
                    products, it may have plural Category MIB
                    modules."
          ::= { proV1AgentInfo 2 }


proV1AgentMIBCatEntry OBJECT-TYPE
          SYNTAX ProV1AgentMIBCatEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
                    "An entry in the proV1AgentMIBCatTable."
          INDEX {
                    proV1AgentMIBCatIdx }
          ::= { proV1AgentMIBCatTable 1 }


ProV1AgentMIBCatEntry ::= SEQUENCE {

          proV1AgentMIBCatIdx Integer32,
          proV1AgentMIBCatOID OBJECT IDENTIFIER }


proV1AgentMIBCatIdx OBJECT-TYPE
          SYNTAX Integer32 (1..2147483647)
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "This is an index number for specifying the
                    entry in this table."
          ::= { proV1AgentMIBCatEntry 1 }


proV1AgentMIBCatOID OBJECT-TYPE
          SYNTAX OBJECT IDENTIFIER
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "This is the value of the object identifier
                    for the MIB module that the equipment (Agent)
                    holds."
          ::= { proV1AgentMIBCatEntry 2 }
```

```
proV1AgentMIBProdTable OBJECT-TYPE
          SYNTAX SEQUENCE OF ProV1AgentMIBProdEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
                    "This table shows which expansion MIB is mounted
                    in the SNMP agent in addition to the Common
                    MIB. This table contains information on
                    expanded MIB for each product. The top object
                    identifier of the mounted Product MIB module
                    is held in this table. In the case of complex
                    product, it may have multiple Product MIB
                    modules. The structure of this table is the
                    same as the structure of the Category MIB
                    table."
          ::= { proV1AgentInfo 3 }


proV1AgentMIBProdEntry OBJECT-TYPE
          SYNTAX ProV1AgentMIBProdEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
                    "An entry to proV1AgentMIBProdTable."
          INDEX {
                    proV1AgentMIBProdIdx }
          ::= { proV1AgentMIBProdTable 1 }


ProV1AgentMIBProdEntry ::= SEQUENCE {

          proV1AgentMIBProdIdx Integer32,
          proV1AgentMIBProdOID OBJECT IDENTIFIER }


proV1AgentMIBProdIdx OBJECT-TYPE
          SYNTAX Integer32 (1..2147483647)
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "This is an Index number for specifying the
                    entry in this table."
          ::= { proV1AgentMIBProdEntry 1 }


proV1AgentMIBProdOID OBJECT-TYPE
          SYNTAX OBJECT IDENTIFIER
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "This is the value of the object identifier
                    for the MIB module that the equipment
                    (Agent) holds."
          ::= { proV1AgentMIBProdEntry 2 }
```

```
proV1TrapDestTable OBJECT-TYPE
          SYNTAX SEQUENCE OF ProV1TrapDestEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
                    "The trap destination address (IP address) for
                    SNMP agent is stored. The trap destination
                    address is generally set up manually using the
                    control panel, web browser, telnet, or the like
                    when the product is installed. For some products,
                    it is desirable to automate this process and
                    simplify the installing operation. For such
                    products, the structure takes into account
                    setting up the trap destination address
                    automatically through negotiation with the
                    SNMP Manager."
          ::= { proV1AgentInfo 4 }


proV1TrapDestEntry OBJECT-TYPE
          SYNTAX ProV1TrapDestEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
                    "Entry to the proV1TrapDestTable"
          INDEX {
                    proV1TrapDestIdx }
          ::= { proV1TrapDestTable 1 }


ProV1TrapDestEntry ::= SEQUENCE {

          proV1TrapDestIdx         Integer32,
          proV1TrapDestAddress     IpAddress,
          proV1TrapDestEntryStatus INTEGER }


proV1TrapDestIdx OBJECT-TYPE
          SYNTAX Integer32 (1..2147483647)
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "This is an index number for specifying the
                    entry in this table."
          ::= { proV1TrapDestEntry 1 }


proV1TrapDestAddress OBJECT-TYPE
          SYNTAX IpAddress
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
```

```
                    "The trap destination address (IP Address)  is
                    displayed."
          ::= { proV1TrapDestEntry 2 }


proV1TrapDestEntryStatus OBJECT-TYPE
          SYNTAX INTEGER {
                              valid(1),
                              createRequest(2),
                              underCreation(3),
                              invalid(4),
                              fixed(5) }
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
                    "This variable is for adding entries from the
                    SNMP Manager. The status 5 (fixed) has been
                    added to the definition of the Entry Status
                    generally used in network system industry for
                    managing table format of the MIB. 1-valid,
                    2-createRequest, 3-underCreation, 4-invalid,
                    5-fixed."
          ::= { proV1TrapDestEntry 3 }


proV1Reserved OBJECT-IDENTITY
          STATUS current
          DESCRIPTION ""
          ::= { proV1AgentInfo 5 }

proV1LocalClockTime OBJECT-TYPE
          SYNTAX ProV1TimeStamp (SIZE (32))
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "When the unit has a built-in clock, its time
                    is returned. In case the product does not have
                    year, month, day or time information '9999'
                    or '99' should be inserted in each column."
          ::= { proV1AgentInfo 6 }


proV1ErrorStatus OBJECT-IDENTITY
          STATUS current
          DESCRIPTION
                    "Information on the errors and warnings
                    currently generated in the product shall be
                    stored in table format. This is not a table
                    storing a history of past error/warning
                    information. When the error/warning is ended,
                    its information is deleted from the table. At
                    that time, information of the end of
                    error/warning may be temporarily displayed on
                    the table."
```

```
          ::= { proV1Common 3 }

proV1ErrStatTable OBJECT-TYPE
          SYNTAX SEQUENCE OF ProV1ErrStatEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
                    "Information on the errors and warnings
                    currently generated in the product."
          ::= { proV1ErrorStatus 1 }


proV1ErrStatEntry OBJECT-TYPE
          SYNTAX ProV1ErrStatEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
                    "Entry to the proV1ErrorStatusTable."
          INDEX {
                    proV1ErrStatProdId,
                    proV1ErrStatErrIdx }
          ::= { proV1ErrStatTable 1 }


ProV1ErrStatEntry ::= SEQUENCE {

          proV1ErrStatProdId Integer32,
          proV1ErrStatErrIdx Integer32,
          proV1ErrStatLevel  INTEGER,
          proV1ErrStatCode   DisplayString,
          proV1ErrStatDescr  DisplayString,
          proV1ErrStatTime   ProV1TimeStamp }


proV1ErrStatProdId OBJECT-TYPE
          SYNTAX Integer32 (1..2147483647)
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "The index number of the Product ID Information
                    Table shall be entered to specify the product
                    that the error/warning is generated in. If the
                    product is a stand-alone product and has no
                    other product underneath it, 1 is entered as
                    the index value."
          ::= { proV1ErrStatEntry 1 }


proV1ErrStatErrIdx OBJECT-TYPE
          SYNTAX Integer32 (1..2147483647)
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "When managing faults within the product, an
```

```
                    index number to specify the fault shall be
                    entered. For example, when three faults are
                    detected, the corresponding error index
                    numbers of them are attached in order to manage
                    and transmit the fault information."
          ::= { proV1ErrStatEntry 2 }


proV1ErrStatLevel OBJECT-TYPE
          SYNTAX INTEGER {
                              critical(1),
                              major(2),
                              minor(3),
                              warning(4),
                              clear(5),
                              information(6) }
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "A value should be entered showing how critical
                    the error is. 1-critical,  2-major,
                    3-minor,  4-warning,  5-clear(Normal),
                    6-information"
          ::= { proV1ErrStatEntry 3 }


proV1ErrStatCode OBJECT-TYPE
          SYNTAX DisplayString (SIZE (0..64))
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "An error code shall be entered that is
                    prescribed for the product and product category.
                    Alphanumeric characters can be used, but
                    capital letters and small letters are not
                    distinguished. Also, do not use symbols. Also,
                    because the codes take the form of character
                    strings not numbers, the same numeric value
                    written in a different way is recognized as
                    a different code, for example 0001 and 1."
          ::= { proV1ErrStatEntry 4 }


proV1ErrStatDescr OBJECT-TYPE
          SYNTAX DisplayString (SIZE (0..255))
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "An explanation of the error or error name shall
                    be entered with which the gist of the meaning
                    of the error/warning can be understood."
          ::= { proV1ErrStatEntry 5 }
```

```
proV1ErrStatTime OBJECT-TYPE
          SYNTAX ProV1TimeStamp (SIZE (32))
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "The time at which the error/warning (or the
                    end of error/warning) was generated shall be
                    entered (yyyy/mm/dd hh:mm:ss:ff GMT+xx:xx).
                    Here, the ff expresses the frame number and
                    the GMT+xx:xx expresses the time difference
                    from Greenwich Mean Time (GMT). In case of the
                    product which does not have year, month, day
                    or time information, '9999' or '99' should be
                    filled in each column."
          ::= { proV1ErrStatEntry 6 }


proV1ErrSttEqSttTable OBJECT-TYPE
          SYNTAX SEQUENCE OF ProV1ErrSttEqSttEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
                    "Information on the equipment alarms currently
                    generated in the product is stored in table
                    format. This is not a table storing a history
                    of past equipment alarm information."
          ::= { proV1ErrorStatus 2 }


proV1ErrSttEqSttEntry OBJECT-TYPE
          SYNTAX ProV1ErrSttEqSttEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
                    "Entry to the proV1ErrSttEqSttTable"
          INDEX {
                    proV1ErrSttEqSttProdId }
          ::= { proV1ErrSttEqSttTable 1 }


ProV1ErrSttEqSttEntry ::= SEQUENCE {

          proV1ErrSttEqSttProdId   Integer32,
          proV1ErrSttEqSttAlmLevel INTEGER,
          proV1ErrSttEqSttOprStt   INTEGER,
          proV1ErrSttEqSttUsgStt   INTEGER }


proV1ErrSttEqSttProdId OBJECT-TYPE
          SYNTAX Integer32 (1..2147483647)
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "The index number of the Product Information
```

```
                    Table shall be entered to specify the product
                    that the equipment status is generated in. If
                    the product is a stand-alone product and has
                    no other product underneath it, 1 is entered
                    as the index value."
          ::= { proV1ErrSttEqSttEntry 1 }


proV1ErrSttEqSttAlmLevel OBJECT-TYPE
          SYNTAX INTEGER {
                              critical(1),
                              major(2),
                              minor(3),
                              warning(4),
                              clear(5),
                              information(6) }
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "A value shall be entered showing how critical
                    the alarm is. 1-critical, 2-major, 3-minor,
                    4-warning, 5-normal."
          ::= { proV1ErrSttEqSttEntry 2 }


proV1ErrSttEqSttOprStt OBJECT-TYPE
          SYNTAX INTEGER {
                              disabled(1),
                              enabled(2) }
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "The operation state of the equipment shall
                    be entered. The values are classified into 2
                    states as follows. Meaning of each state
                    conforms to ITU-T X.731. 1-disabled,
                    2-enabled"
          ::= { proV1ErrSttEqSttEntry 3 }


proV1ErrSttEqSttUsgStt OBJECT-TYPE
          SYNTAX INTEGER {
                              idle(1),
                              active(2),
                              busy(3) }
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                    "The usage state of the equipment shall be
                    entered. The value is classified into 3 states
                    as follows. The meaning of each state conforms
                    to ITU-T X.731. 1-idle, 2-active, 3-busy"
          ::= { proV1ErrSttEqSttEntry 4 }
```

```
proV1Trap OBJECT-IDENTITY
          STATUS current
          DESCRIPTION
                    "The trap announcing the occurrence (and end)
                    of error/warning is defined as collections of
                    the items."
          ::= { proAVMIB1 5 }

proV1CommonConformance OBJECT IDENTIFIER          ::= { proV1Common 4 }

proV1Compliances OBJECT IDENTIFIER ::= { proV1CommonConformance 1 }

proV1Groups OBJECT IDENTIFIER  ::= { proV1CommonConformance 2 }


--
-- Trap information (Specific Trap No. = 65793 (0001 0000 0001 0000 0001)
--

proV1Trap1 NOTIFICATION-TYPE
          OBJECTS {
                    proV1ProdIdInfoIdx,
                    proV1ProdIdInfoDevID,
                    proV1ProdIdInfoManufact,
                    proV1ProdIdInfoModel,
                    proV1ProdIdInfoDest,
                    proV1ProdIdInfoSN,
                    proV1ProdIdInfoCat,
                    proV1ProdIdInfoVer,
                    proV1ProdIdInfoModelSuffix,
                    proV1ProdIdInfoVerSuffix,
                    proV1ErrStatErrIdx,
                    proV1ErrStatLevel,
                    proV1ErrStatCode,
                    proV1ErrStatDescr,
                    proV1ErrStatTime}
          STATUS current
          DESCRIPTION
                    "This is trap collection to notify of Error
                    / Warning (or recover)."
          ::= { proV1Trap 65793 }

proV1ProductInfoGroup OBJECT-GROUP
          OBJECTS {
                    proV1ProdIdInfoIdx,
                    proV1ProdIdInfoDevID,
                    proV1ProdIdInfoManufact,
                    proV1ProdIdInfoModel,
                    proV1ProdIdInfoDest,
                    proV1ProdIdInfoSN,
                    proV1ProdIdInfoCat,
                    proV1ProdIdInfoVer,
                    proV1ProdIdInfoModelSuffix,
```

```
                    proV1ProdIdInfoVerSuffix }
          STATUS current
          DESCRIPTION ""
          ::= { proV1Groups 1 }

proV1ModuleGroup OBJECT-GROUP
          OBJECTS {
                    proV1ModuleProdId,
                    proV1ModuleIdx,
                    proV1ModuleType,
                    proV1ModuleName,
                    proV1ModuleVer,
                    proV1ModuleNameSuffix,
                    proV1ModuleVerSuffix }
          STATUS current
          DESCRIPTION ""
          ::= { proV1Groups 2 }

proV1AgentInfoGroup OBJECT-GROUP
          OBJECTS {
                    proV1AgentMIBVer,
                    proV1AgentMIBCatIdx,
                    proV1AgentMIBCatOID,
                    proV1AgentMIBProdIdx,
                    proV1AgentMIBProdOID,
                    proV1LocalClockTime }
          STATUS current
          DESCRIPTION ""
          ::= { proV1Groups 3 }

proV1ErrorStatusGroup OBJECT-GROUP
          OBJECTS {
                    proV1ErrStatProdId,
                    proV1ErrStatErrIdx,
                    proV1ErrStatLevel,
                    proV1ErrStatCode,
                    proV1ErrStatDescr,
                    proV1ErrStatTime,
                    proV1ErrSttEqSttProdId,
                    proV1ErrSttEqSttAlmLevel,
                    proV1ErrSttEqSttOprStt,
                    proV1ErrSttEqSttUsgStt }
          STATUS current
          DESCRIPTION ""
          ::= { proV1Groups 4 }

proV1TrapGroup OBJECT-GROUP
          OBJECTS {
                    proV1TrapDestIdx,
                    proV1TrapDestAddress,
                    proV1TrapDestEntryStatus }
          STATUS current
          DESCRIPTION ""
          ::= { proV1Groups 5 }
```

```
proV1NotificationGroup NOTIFICATION-GROUP
          NOTIFICATIONS {
                    proV1Trap1 }
          STATUS current
          DESCRIPTION ""
          ::= { proV1Groups 6 }


proV1Compliance MODULE-COMPLIANCE
          STATUS obsolete
          DESCRIPTION ""

          MODULE
          MANDATORY-GROUPS {
                              proV1ProductInfoGroup,
                              proV1ModuleGroup,
                              proV1AgentInfoGroup,
                              proV1ErrorStatusGroup,
                              proV1TrapGroup,
                              proV1NotificationGroup }

          ::= { proV1Compliances 1 }

END
```