Printer Management Interface

Common MIB/MIF Specification

Version 0.1, May 15, 1994

Files: prtspec1.dcr, prtspec1.psr

By

Ronald L. Smith

Texas Instruments

email: rlsmith@nb.ppd.ti.com
phone: (817) 774-6151

F.D. Wright

Lexmark International

email: don@lexmark.com

phone: 606-232-4808

Thomas N. Hastings

Digital Equipment Corp.

email: hastings@hannah.enet.dec.com

phone: (508) 467-8299

Stephen N. Zilles

Adobe Systems, Inc

email: szilles@mv.us.adobe.com

phone: (415) 962-4766

Joel Gyllenskog

Hewlett Packard

email: jgyllens@hpdmd48.boi.hp.com

phone: (208) 232-4515

This document is the complete specification of the semantics of the IETF Printer MIB and the DMTF Printer MIF in a manner that is independent of the syntax for MIB and MIF specifications. After this document is relatively stable, it will be converted into the appropriate MIB Internet Draft and MIF format specifications, using the exact text from this document as a starting point and adding the appropriate MIB and MIF syntax.

1. INTRODUCTION 2
1.1. Network Printing Environment 2
1.2. Printer Device Overview 3
1.3. Categories of Printer Information 3
1.3.1. Descriptions 3
1.3.2. Status 3
1.3.3. Alerts 3
2. PRINTER MODEL 4
2.1. Overview of the Printer Model 5
2.2. Printer Sub-Units 5
2.2.1. General Printer 5
2.2.2. Inputs 5
2.2.3. Media 5
2.2.4. Outputs 5
2.2.5. Finishers 6
2.2.6. Markers 6
2.2.7. Media Paths 6
2.2.8. System Controller 6
2.2.9. Interfaces 7
2.2.10. Channels 7
2.2.11. Interpreters 7
2.2.12. Console 7
2.2.13. Alerts 7
2.2.13.1. Status and Alerts 7
2.2.13.2. Overall Printer Status 7
2.2.13.2.1. Host MIB Printer Status 8
2.2.13.2.2. Sub-unit Status 10
2.2.13.3. Alert Tables 10
2.2.13.4. Alert Table Management 11
2.3. Read-Write Objects/Attributes 11
2.4. Notational Conventions 12
2.4.1. Object/Attribute Names 12
2.4.1.1. SNMP MIB Object Names 12
2.4.1.2. DMTF MIF Attribute Names 13
2.4.2. Defined Data Types 13
2.4.3. Enumerations 13
2.4.3.1. SNMP MIB enumeration symbols 13
2.4.3.2. DMTF MIF enumeration symbols 13
2.4.3.3. Registering Additional Enumerated Values 13
3. OBJECTS/ATTRIBUTES FROM OTHER MIB/MIF SPECIFICATIONS 15
3.1. System Group objects 15
3.2. System Controller 15
3.3. Interface Group objects 15
4. GENERAL PRINTER GROUP 16
4.1. Skin Table - Mandatory 16
4.2. Localization Table - Mandatory 16
4.3. CharacterSetTable - Mandatory 16

4.4. General Printer - Extended - Optional 17
5. INPUT GROUP - MANDATORY 18
5.1. Input Table - Mandatory 18
5.2. Input Group Type Extended - Optional 19
6. MEDIA GROUP - OPTIONAL 20
7. OUTPUT GROUP - MANDATORY 22
7.1. Output Table - Mandatory 22
7.2. Output Group - Type Extended Objects - Optional 22
7.3. Output - Dimensions Extended - Optional 23
7.4. Output - Features - Optional 23
8. FINISHER GROUP - OPTIONAL 25
8.1. Output Associated Finishers Table - Optional 25
8.2. Finishing Configuration - Optional 26
8.3. Finisher Table - Optional 26
9. MARKER GROUP - MANDATORY 27
9.1. Marker Table - Mandatory 27
9.2. Marker Supplies Table - Optional 28
9.3. Marker Colorant Table - Optional 28
10. MEDIA PATH GROUP - MANDATORY 30
10.1. Media Path Table - Optional 30
11. CHANNEL GROUP - MANDATORY 31
11.1. The channel table and its underlying structure 31
11.2. Channel Table - Mandatory 31
12. INTERPRETER GROUP - MANDATORY 33
12.1. Interpreter Table - Mandatory 33
13. CONSOLE GROUP - MANDATORY 35
13.1. Display Buffer Table - Mandatory 35
14. ALERTS GROUP - MANDATORY 36
15. ENUMERATED VALUES 38
15.1. Sub-Unit Status 38
15.2. Units of measure 39
15.3. Device Types 39
15.4. Miscellaneous 41
15.5. ChannelType (enum type 2) 41
15.6. IntpLangFam - Interpreter Language Family (enum type 2) 42
15.7. Alert Codes 44
16. APPENDIX A - GLOSSARY OF TERMS 46
17. APPENDIX B - MEDIA SIZE NAMES 48
18. APPENDIX C -MEDIA NAMES 50
19. APPENDIX D - ROLES OF USERS 52
19.1. Background 52
19.2. Proposed Print System Requirements Areas 52
19.3. Proposed Role Models 53
19.4. Matrix of Requirement Areas and Role Models 53
20. APPENDIX E - PARTICIPANTS 55
21. APPENDIX Y - ISSUES 56
22. APPENDIX Z - CHANGE HISTORY 59
22.1. Changes from Version 0.7, 27-Apr to Version 0.1, 15-May 59

22.2. Changes from Version 0.6, 26-Apr, to Version 0.7, 27-Apr 60
22.3. Changes from Version 0.2, 25-Apr, to Version 0.6, 26-Apr 60
22.4. Changes from Version 0.1, 23-Apr, to Version 0.2, 25-Apr 60
22.5. Changes to create Version 0.1, 23-Apr 61
23. INDEX **62**

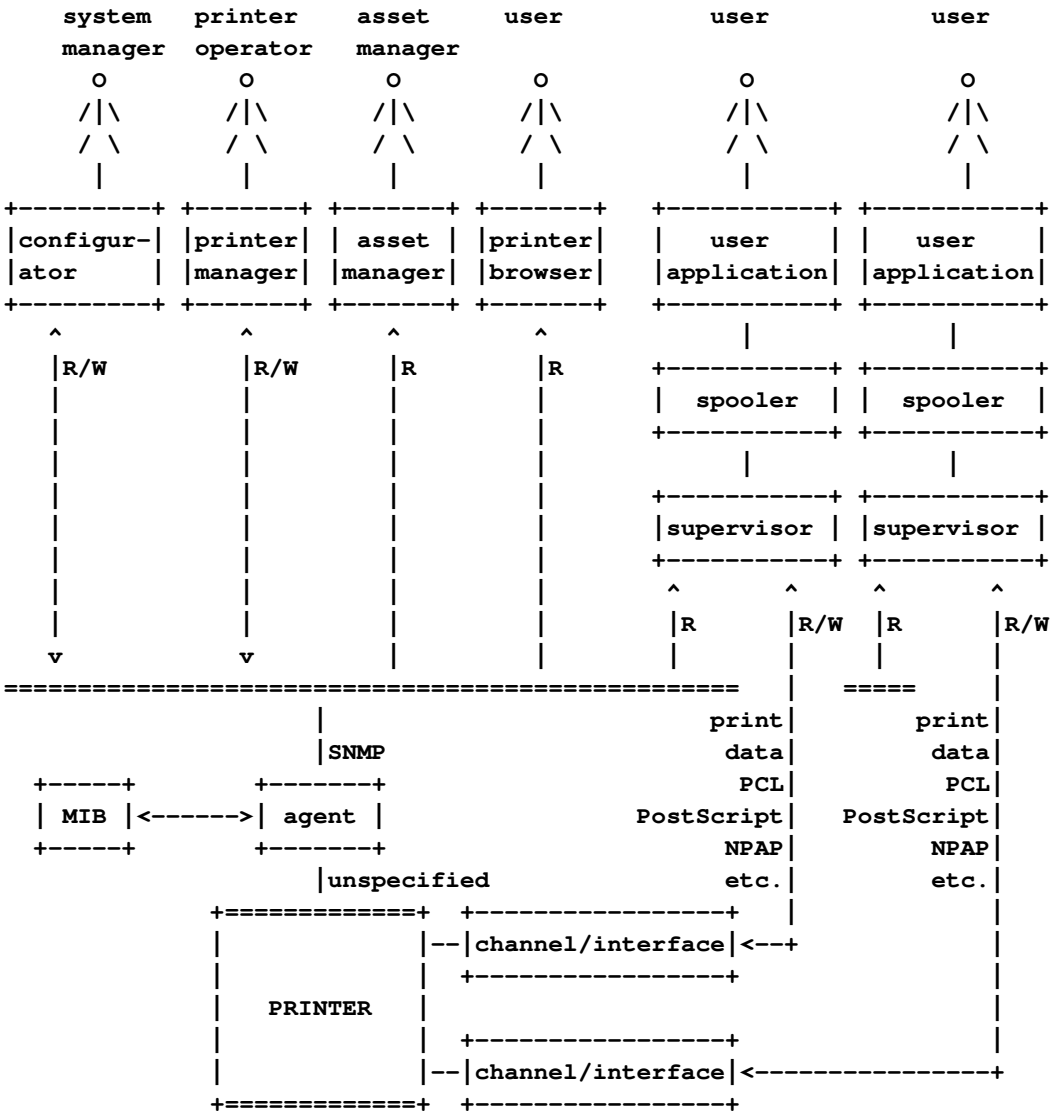
1. Introduction

The Printer Management Interface (PMI) Common MIB/MIF Specification specifies the semantics of the IETF Printer MIB and the DMTF Printer MIF in a manner that is independent of the syntax for MIB and MIF specifications. After this document is relatively stable, it will be converted into the appropriate MIB Internet Draft and MIF format specifications, using the exact text from this document as a starting point and adding the appropriate MIB and MIF syntax.

1.1. Network Printing Environment

The management of producing a printed document, in any computer environment, is a complex subject. Basically, the task can be divided into two overlapping pieces, the management of printing and the management of the printer. Printing encompasses the entire process of producing a printed document from generation of the file to be printed, selection of a printer, choosing printing properties, routing, queuing, resource management, scheduling, and final printing including notifying the user. Most of the printing process is outside the scope of the model presented here; only the management of the printer is covered.

Figure 1 - One Printer's View of the Network



1.2. Printer Device Overview

A printer is the physical device that takes media from an input source, produces marks on that media according to some page description or page control language and puts the result in some output destination, possibly with finishing applied. Printers are complex devices that consume supplies, produce waste and have mechanical problems. In the management of the physical printing device the description, status and alert information concerning the printer and its various subparts has to be made available to the management application so that it can be reported to the end user, key operators for the replenishment of supplies or the repair or maintenance of the device. The information needed in the management of the physical printer and the management of a printing job highly overlap and many of the tasks in each management area require the same or similar information .

1.3. Categories of Printer Information

Information about printers is classified into three basic categories, descriptions, status and alerts.

1.3.1. Descriptions

Descriptions convey information about the configuration and capabilities of the printer and its various sub-units. This information is largely static information and does not generally change during the operation of the system but may change as the printer is repaired, re-configured or upgraded. The descriptions are one part of the visible state of the printer where *state* means the condition of being of the printer at any point in time.

1.3.2. Status

Status is the information regarding the current operating state of the printer and its various sub-units. Status is the rest of the visible state of the printer. As an example of the use of status, a management application must be able to determine if the various sub-units are ready to print or are in some state that prevents printing or may prevent printing in the future.

1.3.3. Alerts

An Alert is the representation of a reportable event in the Printer. An event is a change in the state of the printer. Some of those state changes are of interest to a management application and are therefore reportable. Typically, these are the events that affect the printer's ability to print. Alerts usually occur asynchronously to the operation of the computer system(s) to which the printer is attached. In principle, all Alerts are redundant because the management application could check the entire visible state of the printer and compare it to a previously saved version of that state, thereby detecting changes of state. Alerts provide a more convenient indication of a state change to management applications. For convenience below, "alert" will be used for both the event caused by a change in the printer's state and for the representation of that event.

Alerts can be classified into two basic categories, critical and non-critical. A critical alert is one that is triggered by entry into a state in which the printer is stopped and printing can not continue until the condition that caused critical alert is eliminated. "Out of paper", "toner empty" and "output bin" full are examples of critical alerts. Non-critical alerts are triggered by those events that enter a state in which printing is not stopped. Such a non-critical state may, at some future time, lead to a state in which printing may be stopped. Examples of this kind of non-critical alerts are "input media low", "toner low" and "output bin nearly full". Or, a non-critical alert may simply provide information, such as signaling a configuration changed in the printer.

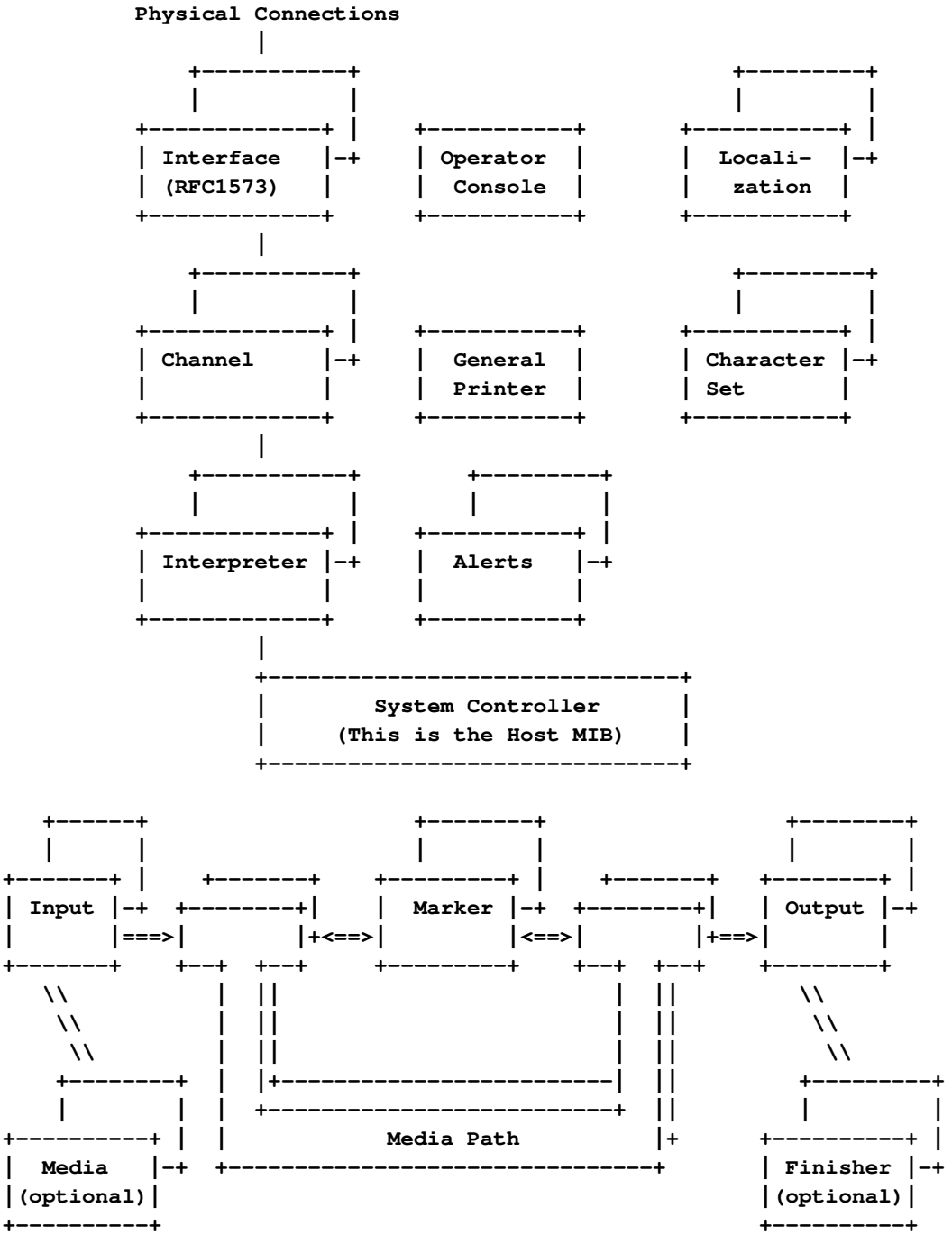
Description, status and alert information about printer can be thought of as a data base describing the printer. The management application for a printer will want to view the printer data base differently depending on how and for what purposes the information in the data base is needed.

2. Printer Model

In order to accomplishment the management of the printer, an abstract model of the printer is needed to represent the sub-units from which the printer is composed. A printer can be described as consisting of 16 types of sub-units. It is important to note that the sub-units of a printer do not necessarily relate directly to any physically identifiable mechanism. Sub-units can also be a set of definable logical processes, such as interpreters for page description languages or command processors that set various operating modes of the printer.

Figure 2 shows a block diagram of the printer and its basic 16 sub-units.

Figure 2 - Printer Block Diagram



2.1. Overview of the Printer Model

The model has three basic parts: (1) the flow of a print file into an interpreter and onto the marker, (2) the flow of media through the marker and (3) the auxiliary sub-units that control and facilitate the two prior flows. The flow of the print data comes through a physical connection on which some form of transport protocol stack is running. The data provided by the transport protocol (interface) appears on a channel which is the input to an interpreter. The interpreter converts the print data into a form suitable for marking on the media.

The media resides in Input sub-units from which the media is selected and then transported via a Media Path first to a Marking sub-unit and then onto an Output sub-unit with (optionally) some finishing operations being performed. The auxiliary sub-units facilitate control of the printer, inquiry/control of the operator panel, reporting of alerts, and the adaptation of the printer to various natural languages and characters sets. All the software sub-units run on the System Controller which represents the processor, memory and storage systems of the Printer. Each of the sub-units is discussed in more detail below.

All of the sub-units other than the Alerts report only state information, either a description or a status. The Alerts sub-unit reports event information.

2.2. Printer Sub-Units

A printer is composed of 16 types of sub-units, called groups. The following sections describe the different types of sub-units.

2.2.1. General Printer

The general printer sub-unit is responsible for the overall control and status of the printer. There is exactly one general printer sub-unit in a printer. The general printer sub-unit is represented by the General Printer Group in the model. In addition to the providing the status of the whole printer and allowing the printer to be reset, this Group provides information on the status of the packaging of the printer, in particular, the doors. The general printer sub-unit is usually implemented on the system controller.

The localization portion of the general printer sub-unit is responsible for identifying the natural language, country, and character set in which character strings are expressed. There may be one or more localizations supported per printer. The available localizations are represented by the Localization table.

The character set portion of the general printer sub-unit is responsible for identifying the possible character sets that are used by the interpreters and the operator console. There may be one or more character sets per printer. The understood character sets are represented by the Character Set Table.

2.2.2. Inputs

Input sub-units are mechanisms that feed media to be marked on into the printer. A printer contains one or more input sub-units. These are represented by the Input Group in the model. The model does not distinguish fixed input bins from removable trays, except to report when a removable tray has been removed.

There are as many input sub-units as there are distinctly selectable input "addresses". For example, if a tray has an option for manually feeding paper as well as automatically feeding from the tray, then this is two input sub-units if these two sources can be (must be) separately selected and is one input sub-unit if putting a sheet in the manual feed slot overrides feeding from the contents of the tray; that is, in the second case there is no way to separately select or address the manual feed slot.

2.2.3. Media

An input sub-unit can hold one or more instances of the media on which marking is to be done. Typically, there is a large set of possible media that can be associated with an input. The Media Group is an extension of the Input Group which represents that media that is in an input sub-unit. The Media Group only describes that current contents of each input and not the possible content of the input sub-unit.

2.2.4. Outputs

Output sub-units are mechanisms that receive media that has been marked on. A printer contains one or more output mechanisms. These are represented by the Output Group in the model. The model does not distinguish fixed output bins from removable output bins, except to report when a removable bin has been removed.

There are as many output sub-units as there are distinctly selectable output "addresses". Output sub-units can be addressed in two different ways: (1) as a set of "mailboxes" which addressed by a specific mailbox selector such as a bin number or a bin name, or (2) as a set of "slots" into which multiple copies are collated. Sometimes both modes of using the output sub-units can be used on the same printer. All that is important from the viewpoint of the model is that the output units can be separately selected.

2.2.5. Finishers

A finisher is a sub-unit that performs some operation on the media other than marking. The finisher sub-units are represented by the Finisher Group in the model. Some examples of finishing processes are stapling, punching binding, inserting or folding. Finishing processes may have supplies associated with the process. Stapling, binding and punching are examples of processes that have supplies. A printer may have more than one finishing device and each finishing device may be associated with one or more output devices.

The exact interaction and sequencing between an output device and its associated finisher is not specified by the model. It depends on the type of finishing process and the exact implementation of the printer system. This standard allows for the logical association of a finishing process with an output device but does not put any restrictions on the exact sequence or interaction with the associated output device. The output and finisher sub-units may or may not be separate identifiable physical mechanisms depending on the exact implementation of a printer. In addition, a single output device may be associated with multiple finishing sub-units and a single finishing sub-unit may be associated with multiple output devices.

2.2.6. Markers

A marker is the mechanisms that produces marks on the print media. The marker sub-units and their associated supplies are represented by the Marker Group in the model. A printer can contain one or more marking mechanisms. Some examples of multiple marker sub-units are: a printer with separate markers for normal and magnetic ink or an imagesetter that can output to both a proofing device and final film. Each marking device can have its own set of characteristics associated with it, such as marking technology and resolution.

In this model the marker sub-unit is viewed as very generalized and encompasses all aspects of a marking process. For example, in a xerographic process, the marking process as well as the fusing process would be included in the generalized concept of the marker. With the generalized concept of a marking process, the concept of multiple marking supplies associated with a single marking sub-unit results. For example, in the xerographic process, there is not only a supply of toner, but there can also be other supplies such as a fuser supply that can be consumed and replaced separately. In addition there can be multiple supplies of toner for a single marker device, as in a color process.

2.2.7. Media Paths

The media paths encompass the mechanisms in the printer that move the media through the printer and connect all other media related sub-units: inputs, outputs, markers and finishers. A printer contains one or more media paths. These are represented by the Media Path Group in the model. The Media Path group has some attributes that apply to all paths plus a table of the separate media paths.

In general, the design of the media paths determines the maximum speed of the printer as well as the maximum media size that the printer can handle. Media paths are complex mechanisms and can contain many different identifiable sub-mechanisms such as media movement devices, media buffers, duplexing units and interlocks. Not all of the various sub-mechanisms reside on every media path. For example, one media path may provide printing only on one surface of the media (a simplex path) and another media path may have a sub-mechanism that turns the media over and feeds it a second time through the marker sub-unit (a duplex path). The duplex path may even have a buffer sub-mechanism that allows multiple copies of the obverse side to be held before the reverse side of all the copies is marked.

2.2.8. System Controller

The System Controller is the sub-unit upon which the software components of the Printer run. The System Controller is represented in the model by the Host MIB. This MIB allows for the specification of the processor(s), memory, disk storage, file system and other underlying sub-mechanisms of the printer. The controller can range from simple single processor systems to multiprocessor systems. In addition, controllers can have a full range of resources such as hard disks. The printer is modeled to have one system controller even though it may have more than one processor and multiple other resources associated with it.

2.2.9. Interfaces

An interface is the communications port and associated protocols that are responsible for the transport of data to the printer. A printer has one or more interface sub-units. The interfaces are represented by the Interfaces Group of MIB-II as extended by RFC 1573. Some examples of interfaces are serial ports (with little or no protocol) and EtherNet ports on which one might run InterNet IP, Novell IPX, etc.

2.2.10. Channels

The channel sub-units identify the independent sources of print data (here print data is the information that is used to construct printed pages and may have both data and control aspects). A printer may have one or more channels. The channel sub-units are represented by the Channel Group in the Model. Each channel is typically identified by the electronic path and service protocol used to deliver print data to the printer. A channel sub-unit may be independently enabled (allowing print data to flow) or disabled (stopping the flow of print data). It has a current Control Language which can be used to specify which interpreter is to be used for the print data and to query and change environment variables used by the interpreters (and SNMP). There is also a default interpreter that is to be used if an interpreter is not explicitly specified using the Control Language. Channel sub-units are based on an underlying interface.

2.2.11. Interpreters

The interpreter sub-units are responsible for the conversion of a description of intended print instances into images that are to be marked on the media. A printer may have one or more interpreters. The interpreter sub-units are represented by the Interpreter Group in the Model. Each interpreter is generally implemented with software running on the System Controller sub-unit. The Interpreter Table has one entry per interpreter where the interpreters include both Page Description Language (PDL) Interpreters and Control Language Interpreters.

2.2.12. Console

Many printers have a console on the printer, the operator console, that is used to display and modify the state of the printer. The console can be as simple as a few indicators and switches or as complicated as full screen displays and keyboards. There can be at most one such console. This console sub-unit is represented by the Console Group in the model. Although most of the information displayed there is also available in the state of the printer as represented by the various Groups, it is useful to be able to query and modify the operator console remotely. For example, a management application might like to display to its user the current message on the operator console of the remote printer or the management application user might like to modify the current message on the operators console of the remote printer. As another example, one might have a remote application that puts up a pseudo console on a workstation screen. Since the rules by which the printer state is mapped onto the console and vice versa are not standardized, it is not possible to reproduce the console state or the action of console buttons and menus. Therefore, the Console Group provides access to the console. The operator console is usually implemented on the system controller with additional hardware for input and display.

2.2.13. Alerts

The alert sub-unit is responsible for detecting reportable events, making an entry in the alert table and, if and only if the event is a critical event, initiating a trap. The alert sub-unit is represented by the Alerts Group and, in particular, the Alert Table. This table contains information on the severity, sub-unit, detailed location within the sub-unit, alert code and description of each critical alert that is currently active within the printer. Each reportable event causes an entry to be made in the Alert Table.

2.2.13.1. Status and Alerts

Summary information about the state of the printer is reported at three separate levels: (1) there is the status of the printer as a whole reported in the Host MIB, (2) there is the status of various sub-units reported in the principle table of the Group that represents the sub-unit, and (3) there are alert codes reported in the Alert Table.

2.2.13.2. Overall Printer Status

The Printer Working Group postulated a set of status "objects" (quotes used because they did not become SNMP objects) that characterized the overall status of a printer. These named states are:

Non Critical Alert Active - there are one or more Non Critical alerts in the Alert Table. This is the logical or of all of the Non Critical Alert Active bits from all Sub-unit status octets.

Critical Alert Active - There are one or more Critical Alerts in the Alert Table. This the logical or of all of the Critical Alert Active bits from all Sub-unit Status octets.

On Request - The Printer is not currently available to service requests, but could be made available (brought to the ready state) with some human interaction.

Unavailable - The device is unavailable for use (this is the same as "broken" or "down" in other terminology's)

Busy / Temporarily Unavailable - The device is operational but currently occupied with a request for activity. The device will become available without the need of human interaction.

Moving on- or off-line - The device is either off-line, in the process of moving off-line or in the process of moving back on-line; for example on high end printers reloading paper involves a transition to off-line to open the paper bin, it is then filled and, finally, there is a transition back to on-line as the paper bin is repositioned for printing.

Rather than define new status objects to indicate these states, the Printer Working Group recognized that the existing status objects in the Host MIB provided much of the information that was desired. Thus, the overall status of the printer can be determined by reading three objects in the Host MIB: (1) hrDeviceStatus in the entry in the Host MIB hrDeviceTable; (2) hrPrinterStatus in the hrPrinterTable; and (3) hrPrinterDetectedErrorState in the hrPrinterTable. The following table shows the mapping of the postulated printer status into particular values for the three status objects related to Printers in the Host MIB:

Printer Status	hrDeviceStatus	hrPrinterStatus	hrPrinterDetectedErrorState
Normal	running(2)	idle(3)	none set
Busy/Temporarily Unavailable	running(2)	printing(4)	
Non Critical Alert Active	warning(3)	idle(3) or printing(4)	could be: lowPaper, or lowToner
Critical Alert Active	down(5)	other(1)	could be: jammed, or noPaper, or noToner, or doorOpen
On Request	down(5)	other(1)	serviceRequested
Unavailable	down(5)	other(1)	
Moving off-line	warning(3)	idle(3) or printing(4)	offline
Off-line	down(5)	other(1)	offline
Moving on-line	down(5)	warmup(5)	

This mapping does not exactly capture the intent of the Status "objects" above. It is not possible to simultaneously should both Critical and Non-Critical Alerts being active. If a Critical Alert is active any Non-Critical Alerts are masked. But, it is possible to detect the Non-Critical Alerts by scanning the Alert Table. The representation of the On Request status uses serviceRequested with the down(5) value of hrDeviceStatus although the Host MIB indicates that a warning(3) value is thought more appropriate when serviceRequested is set. Moving off-line is represented as a warning(3) (indicating that printing may still be possible) when printing may have already ceased in which case hrPrinterStatus should be idle(3). With these exceptions, the representation in terms of the Host MIB objects is workable. Detailed status is reported in the sub-unit status fields.

2.2.13.2.1. Host MIB Printer Status

For completeness, the definitions of the Printer Status objects of the Host MIB are given below:

```
hrDeviceStatus OBJECT-TYPE
     SYNTAX  INTEGER {
          unknown(1),
          running(2),
          warning(3),
          testing(4),
          down(5)
     }
```

```
            ACCESS  read-only
            STATUS  mandatory
            DESCRIPTION
                  "The current operational state of the device
                  described by this row of the table.  A value
                  unknown(1) indicates that the current state of the
                  device is unknown.  running(2) indicates that the
                  device is up and running and that no unusual error
                  conditions are known.  The warning(3) state
                  indicates that agent has been informed of an
                  unusual error condition by the operational software
                  (e.g., a disk device driver) but that the device is
                  still 'operational'.  An example would be high
                  number of soft errors on a disk.  A value of
                  testing(4), indicates that the device is not
                  available for use because it is in the testing
                  state.  The state of down(5) is used only when the
                  agent has been informed that the device is not
                  available for any use."
            ::= { hrDeviceEntry 5 }

    hrPrinterStatus OBJECT-TYPE
           SYNTAX INTEGER {
               other(1),
               unknown(2),
               idle(3),
               printing(4),
               warmup(5)
           }
           ACCESS read-only
           STATUS mandatory
           DESCRIPTION
                   "The current status of this printer device.  When
                   in the idle(1), printing(2), or warmup(3) state,
                   the corresponding hrDeviceStatus should be
                   running(2) or warning(3).  When in the unknown
                   state, the corresponding hrDeviceStatus should be
                   unknown(1)."
           ::= { hrPrinterEntry 1 }

      hrPrinterDetectedErrorState OBJECT-TYPE
           SYNTAX OCTET STRING
           ACCESS read-only
           STATUS mandatory
           DESCRIPTION
                   "This object represents any error conditions
                   detected by the printer.  The error conditions are
                   encoded as bits in an octet string, with the
                   following definitions:

                        Condition          Bit #    hrDeviceStatus

                        lowPaper           0        warning(3)
                        noPaper            1        down(5)
                        lowToner           2        warning(3)
                        noToner            3        down(5)
```

```
            doorOpen          4        down(5)
            jammed            5        down(5)
            offline           6        down(5)
            serviceRequested  7        warning(3)

            If multiple conditions are currently detected and
            the hrDeviceStatus would not otherwise be
            unknown(1) or testing(4), the hrDeviceStatus shall
            correspond to the worst state of those indicated,
            where down(5) is worse than warning(3) which is
            worse than running(2).

            Bits are numbered starting with the most
            significant bit of the first byte being bit 0, the
            least significant bit of the first byte being bit
            7, the most significant bit of the second byte
            being bit 8, and so on.  A one bit encodes that
            the condition was detected, while a zero bit
            encodes that the condition was not detected.

            This object is useful for alerting an operator to
            specific warning or error conditions that may
            occur, especially those requiring human
            intervention."
        ::= { hrPrinterEntry 2 }
```

2.2.13.2.2. Sub-unit Status

Sub-unit status is reported in the entries of the principle table in the Group that represents the sub-unit. For sub-units that report a status, there is a status column in the table and the value of this column is always the following enumeration:

1 other

2 unknown

3 Ready and available

4 In Critical state

5 In Non-critical, warning state

6 State is a Combination of Critical and Warning

7 On request (intervention required)

8 Unavailable

9 Busy/Temporarily Unavailable

For example, an input (tray) that jammed on the next to the last page may show a status of 6 (both a critical state, jammed, and a non critical state, low paper).

2.2.13.3. Alert Tables

The Alert Group consists of a single table in which all active alters are represented. This section provides an overview of the table and a description of how it is managed. The basic content of the alert table is the severity (critical or non-critical) of the alert, the Group and entry where a state change caused the alert, additional information about the alert (a more detailed location, an alert code and a description), and an indication of the level of training needed to service the alert.

The Alert Table contains some information that is redundant, for example that an event has occurred, and some information that is only represented in the Alert Table, for example the additional information. A single table was used because a single entry in a Group could cause more than one alert, for example paper jams in more than one place in a media path. Associating the additional information with the entry in the affected group would only allow one report where associating the additional information with the alert makes multiple reports possible.

Every time an alert occurs in the printer, the printer makes one or more entries into the Alert Table. The printer determines if an event is to be classified as critical or non-critical. If the severity of the Alert is "critical", the printer sends a trap or event notification to the host indicating that the table has changed. Whether or not a trap is sent, the management application is expected to poll the printer on a regular basis and to read and parse the table to determine what conditions have changed, in order to provide reliable information to the management application user.

2.2.13.4. Alert Table Management

The alert tables are sparsely populated tables. This means the tables will only contain entries of the alerts that are currently active and the number of rows, or entries in the table will be dynamic. More than one event can be added or removed from the event tables at a time depending on the implementation of the printer.

There are basically two kinds of events that produce alerts: binary change events and simple change events. Binary change events come in pairs: the leading edge event and the trailing edge event. The leading edge event enters a state from which there is only one exit; for example, going from running to stopped with a paper jam. The only exit from this state is fixing the paper jam and it is clear when that is accomplished. The trailing edge event is the event which exits the state the was entered by the leading edge event; in the example above fixing the paper jam is the trailing edge event.

It is relatively straightforward to manage binary change events in the Alert Table. Only the leading edge event makes an entry in the alert table. This entry persists in the Alert Table until the trailing edge event occurs at which point this event is signal by the removal of the leading edge event entry in the Alert Table. That is, a trailing edge event does not create an entry; it removes the corresponding leading edge event. With binary events it is possible to compute the maximum number that can occur at the same time and construct an Alert Table that would hold that many events. There would be no possibility of table overflow and no information about outstanding events would be lost.

Unfortunately, there are some events that are not binary changes. This other category of event, the simple change event, is illustrated by the configuration change event. With this kind of event the state of the machine has changed, but to a state which is (often) just as valid as the state that was left and from which no return is necessary. For example, an operator may change the paper that is in the primary input source from letter to legal. At some time in the future the paper may be changed back to letter, but it might be changed to executive instead. This is where the problem occurs. It is not obvious how long to keep simple change event entries in the Alert Table. It they were never removed, the Alert Table would continue to grow indefinitely.

There are several approved solutions to the problem of removing simple change events. One approach is to not put simple change events into the Alert Table. Then, as observed above, the table can be designed to never overflow. The other approved approach is to use a policy for removing alert entries when the table is about to overflow. This policy says that entries for Critical Alerts will not be removed as long as there is an entry for a Non-critical Alert that can be removed. After that, it is up to the vendor to decide which Non-critical Alert entry to remove. If there are no Non-Critical Alert entries to remove, then an entry for a Critical Alert can be removed, again vendor's choice. Note that because the Alert Index is a counter there will be gaps in the values in the table which can be detected by the management application to indicate that the management application may want to re-acquire the Printer state and check for state changes it did not observe in the Alert Table.

2.3. Read-Write Objects/Attributes

Some of the objects/attributes in the printer MIB/MIF report on the existence of or amount of a given resource used with the printer. Some examples of such resources are the size and number of sheets of paper in a paper tray or the existence of certain output options. On some printers there are sensors that allow these resources to be sensed. Other printers, however, lack sensors that can detect (all of) the properties of the resource. Because the printer needs to know of the existence or properties of these resources for the printer to function properly some other way of providing this information is needed. The chosen way to solve this problem is to allow a management application to write into object/attributes which hold the descriptive or existence values for printers that cannot sense the values. Thus many of the objects/attributes in the MIB/MIF are given Read/Write access, but which a printer implementation would only permit a write, if the printer could not sense the value itself. Therefore, many of these Read/Write objects/attributes are in fact writeable depending on implementation.

An object/attribute is given Read/Write access in the Printer MIB/MIF specification if:

1. The object/attribute involves installation of a resource that some printers cannot themselves detect. Therefore, external means are needed to inform the printer of the installation. (Here external means include using the operator console, or remote management application) *and*
2. The printer will behave differently if the installation of the resource is reported than the printer would if the installation were not reported; that is, the object/attribute is not to be used as a place to put information not used by the printer, i.e., *not* a "PostIt". Another way of saying this is that the printer believes that information given it and acts as if the information were true. For example, on a printer that cannot sense the size, if one paper size is loaded, but another size is set into the paper size object/attribute, then the printer will use the size that was set as its current paper size in its imaging and paper handling.

The printer may get hints that it may not know about the existence or properties of certain resources. For example, a paper tray may be removed and re-inserted. When this removal and insertion happens, the printer may either assume that a property, such as the size of paper in the tray, has not changed or the printer may change the value of the associated object/attribute to "unknown", as might be done for the amount of paper in the tray. As long as the printer acts according to the value in the object/attribute either strategy is acceptable.

Following the above principle for determining when an object/attribute should have Read/Write access may cause objects/attributes that would otherwise seem to be only Read-Only to be assigned Read/Write in the MIB/MIF specifications. As an example, consider the Media Dimension Units object/attribute. Paper sizes are specified either using metric units (micrometers) or English units (ten thousandths of an inch). A paper size expressed in metric units does not always convert to an integral number of English units and vice versa. Because paper size is Read/Write and because paper size is given in integer values, it is necessary to specify the correct set of units before setting the paper size. Actually, to insure that the Media Dimensions Units and the size object/attributes are all set consistently, this set of changes must all occur in a single atomic operation (a single SNMP packet or a single DMTF Set operation).

2.4. Notational Conventions

This section describes the notational conventions used in this specification in order to be able to completely specify the common syntax and semantics for the corresponding MIB and MIF specifications. A strong goal is to be able to use the MIF to MIB translator in order to guarantee identical semantics and also because the translator generates meta-files as by-products which help in accessing DMTF objects via SNMP. See DMTF Management Interface Specifications.

2.4.1. Object/Attribute Names

This section specifies the syntax of MIB object names and MIF attribute names and the notation used in the common specification so that the respective MIB and MIF syntaxes are unambiguous.

2.4.1.1. SNMP MIB Object Names

SNMP requires that object names start with a lower case letter followed by uppercase and lowercase letters and digits; no other characters are allowed. By convention, each word in an SNMP object name starts with an uppercase letter, except the first. Words that are acronyms, are all uppercase. All object names defined in the Printer MIB start with the prefix **prt** in order to distinguish them from objects defined in other MIBs and to meet the requirement that the first letter must be lower case. As a further convention, the next word(s) in the object name is the Group name. Examples of Printer MIB object names in the **General**, **Input**, and **MediaPath** groups: **prtGeneralConfigChanges**, **prtInputDefaultID**, **prtInputID, and prtMediaPathMaxSpeed**.

As a notational convention to save space in this specification, the **prt** and the group name prefix will not be shown in the MS-WORD tables defining objects/attributes. Instead the prefix and group name is given at the top left of each MS-WORD table. When a description of an object/attribute refers to another object/attribute, the name of that other object/attribute is given in **boldface**. Such references to another object/attribute include neither the **prt** prefix nor the group name prefix, if the object/attribute is in the same group. In the rare cases that a reference is made to an object in another group, both the **prt** prefix and the group name are included in the reference. For example, the description of the **prtInputDescription** object/attribute in the **Input** group contains a reference to the **prtGeneralCurrentLocalization** object/attribute in the **General** group.

2.4.1.2. DMTF MIF Attribute Names

DMTF allows attribute names to be quoted strings containing any ISO Latin-1 character, including SPACE. These attribute names are used to display to users of the management applications, so there is considerable advantage to including a space between each word in a MIF attribute name. However, in order to make the translation to MIB syntax, the Printer MIF will use no special characters such as -, / etc. (even though the MIF to MIB translator would just eliminate them in the conversion). Therefore, the Printer MIF attribute names will consist of words separated by a SPACE and all starting with an uppercase letter, except the prefix **prt**. The corresponding examples of Printer MIF attribute names in the **General**, **Input**, and **MediaPath** groups are: "**prt General Config Changes"**, "**prt Input Default ID"**, "**prt Input ID", and "prt Media Path Max Speed"**.

NOTES:

1. *But see Issue 30 whether to use some special characters.*
2. *But see Issue 29 as to whether* **prt** *is needed.*

2.4.2. Defined Data Types

Both MIBs and MIFs permit the definition of data types to be used elsewhere. For example, enumerated values that are used in more that one object/attribute will be defined as a data type. SNMP requires that datatype name start with an uppercase letter. The data type names will be mixed case with the first letter of each word being uppercase. Since these data type names are for programmers, and are not seen by end users, the words will contain no spaces. Example of MIB and MIF data types are: **IntpLangFam** and **EntStatus**.

2.4.3. Enumerations

Enumerations (enums) are sets of symbolic values defined for use with one or more objects/attributes. Each enumeration set is assigned a symbolic data type name. The enumerations are listed at the end of this specification.

2.4.3.1. SNMP MIB enumeration symbols

SNMP requires that the syntax of an enumeration (defined using the data type **Integer**) start with a lower case letter followed by uppercase and lowercase letters and digits; no other characters are allowed. By convention, each word in an SNMP enum name starts with an uppercase letter, except the first. Words that are acronyms, are all uppercase. For most enums, there is no special (lowercase) prefix. However, for Interpreter Family and Channel Type for which the name are often proper names or acronyms (e.g., PCL, PostScript, ISO6429), the lower case prefixes: **lang** and **chan**, respectively are used. Examples of the Printer MIB enumerations for the **EntStatus** and **IntpLangFam** data types, respectively, are: **bothCriticalAndNonCritical**, and **langPCL, langPostScript**, and **langISO6429**.

2.4.3.2. DMTF MIF enumeration symbols

Like MIF attribute names, MIF enum symbols can contain any ISO Latin-1 character, including SPACE. However, for the same reasons, the Printer MIF will restrict enum names to words starting with an uppercase letter, including the first word (the MIF to MIB translator will automatically lower case the first letter to follow the SNMP requirements). The corresponding examples of Printer MIF enumerations for the **EntStatus** and **IntpLangFam** data types, respectively, are: "**Both Critical And Non Critical"**, and "**lang PCL", "lang PostScript"**, and **"lang ISO 6429"**.

2.4.3.3. Registering Additional Enumerated Values

This working group has defined several type of enumerations. These enumerations differ in the method employed to control the addition of new enumerations. Throughout this document, references to "enum (n)", where n can be 1, 2 or 3 can be found in the various tables. The definitions of these types of enumerations are:

enum (1) All the values are defined in the Printer MIB/MIF specification (RFC for the Printer MIB). Additional enumerated values require a new RFC.

enum (2) An initial set of values are defined in the Printer MIB/MIF specification. Additional enumerated values are registered after review by this working group. The initial versions of the MIB and MIF forms of this specification will contain the values registered so far. After the MIB and MIF forms of this specification are approved, additional values will be registered through IANA *after* approval by this working group.

enum (3) An initial set of values are defined in the Printer MIB/MIF specification. Additional enumerated values are registered without working group review. The initial versions of the MIB and MIF forms of this specification will contain the values registered so far. After the MIB and MIF forms of this specification are approved, additional values will be registered through IANA *without* approval by this working group.

3. Objects/Attributes from other MIB/MIF Specifications

This section lists the objects/attributes from other IETF MIB specifications and other DMTF MIF specifications that are mandatory for conformance to this Printer MIB/MIF specification.

To be supplied by DMTF to correspond to MIB II

DMTF MIF "Component ID: (From DMI Specification - Draft 4.2 - 12 January 1994)

1.	Manufacturer	String64	Name of the organization that produces this component.
2.	Product	String64	Name of this Printer
3.	Version	String64	
4.	Serial Number	String64	
5.	Installation	Date	Time and Date of the Last Installation
6.	Verify	Integer	Verification Level for this printer

"Asking for the 'Verify' attribute causes the component instrumentation code to perform checks to verify that the component is still in the system and working properly. It should return on of the following values:

Value	Meaning
0	An error occurred; Check status code
1	Component does not exist
2	Verify not supported
4	Component exists, functionally untested
5	Component exists, functionally unknown
6	Component exists, functionally no good
7	Component exists, functionally good

NOTES:

See Issue 32, the above attributes are quire different from the System Group of MIB-II.

3.1. System Group objects

This is place holder to list the System Group objects from MIB-II RFC 1213 that shall be implemented when implementing the Printer MIB/MIF.....

3.2. System Controller

This is a place holder to list the Host Resources objects from RFC 1514 that shall be implemented when implementing the Printer MIB/MIF. Possible groups from which we may choose objects include: System, Storage, Device, Running Software, Running Software Performance, and Installed Software.....

3.3. Interface Group objects

This is a place holder to list the Interface Group objects from RFC 1573 that shall be implemented when implementing the Printer MIB/MIF.....

NOTE

1. *Issue 38 - What objects from the System Group (MIB-II RFC 1213), Host Resources MIB (RFC 1514), and Interfaces Group of MIB-II (RFC 1573) shall be listed in the Printer MIB as being required for conformance to the Printer MIB?*

4. General Printer Group

The general printer sub-unit is responsible for the overall control and status of the printer. There is exactly one general printer sub-unit in a printer.

prtGeneral	**Mandatory**			
Object Name	Access	Type	Value	Description
ConfigChanges	Read-Only	Counter32		Counts configuration changes.
DefaultOrientation	Read-Write	Enum (1)	MediaOrientation	The current orientation default. This value applies to all input, all interpreters, etc. unless overridden by other control. This value is a type 1 enum.
CurrentLocalization	Read-Write	Integer32		Index into the localization table specifying the current language, country and character set for **prt*Xxxx*Description** string values.

Notes:

1. Overall printer status can be determined using the following objects from the host MIB: (hrDevice Status, hrPrinterStatus, and hrPrinterDetectedErrorState.

4.1. Skin Table - Mandatory

The skin portion of the General print sub-unit describes the doors and interlocks of the printer. The Skin Table has an entry for each door and interlock.

prtGeneral	**Mandatory**		**prtSkinTable**	
Object Name	**Access**	**Type**	**Value**	**Description**
ID	Read-Only	Integer32		A unique value used by the printer to identify this skin sub-unit. Although these values may change due to a major reconfiguration of the device (e.g. the addition of new skin components to the printer), values are expected to remain stable across successive printer power cycles.
Description	Read-Only	String(255)		The manufacturer provided skin component name in the localization specified by **CurrentLocalization**.
Status	Read-Only	Enum (2)	SkinStatus	The status of this skin sub-unit. This value is a type 2 enum.

Notes:

1. *Issue 33 - Should we change "Skins" to "Doors" everywhere?*

4.2. Localization Table - Mandatory

The localization portion of the General printer sub-unit is responsible for identifying the natural language, country, and character set in which character strings are expressed. There may be one or more localizations supported per printer. The available localizations are represented by the Localization table.

prtGeneral	**Mandatory**		**prtLocalizationTable**	
Object Name	**Access**	**Type**	**Value**	**Description**
Language	Read-Only	String(2)		A two character language code from ISO 639. Examples: EN, FR, DE, ...
Country	Read-Only	String(2)		A two character country code from ISO 3166, a blank string (two space characters) shall indicate that the country is not defined. Examples: US, GB, CA, FR, DE, ...
CharacterSetIndex	Read-Only	Integer32		An index into the character set table identifying the character set used for this localization.

Notes:

1. Added **Index** to CharacterSet name, so that it is different than the next object

4.3. CharacterSetTable - Mandatory

The character set portion of the general printer sub-unit is responsible for identifying the possible character sets that are used by the interpreters and the operator console. There may be one or more character sets per printer. The understood character sets are represented by the Character Set Table.

prtGeneral	**Mandatory**		**prtCharacterSetTable**	
Object Name	Access	Type	Value	Description
CharacterSet	Read-Only	String(63)		A character set name or identifier that identifies this character set.

Notes:

4.4. General Printer - Extended - Optional

prtGeneral - Extended	**Optional**			
Object Name	Access	Type	Value	Description
Reset	Write-Only	Enum (3)	ResetType	Writing causes a reset. This value is a type 3 enum.

Notes:

5. Input Group - Mandatory

Input sub-units are managed as a tabular, indexed collection of possible devices capable of providing media for input to the printing process. Input sub-units typically have a location, a type, an identifier, a set of constraints on possible media sizes and potentially other media characteristics, and may be capable of indicating current status or capacity.

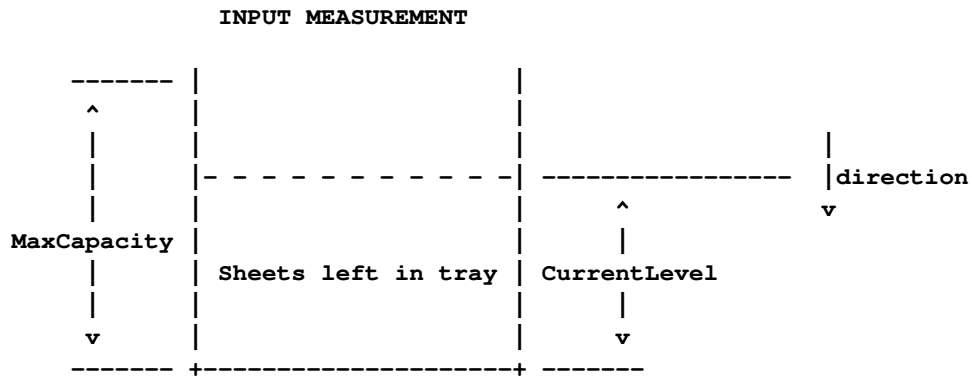
prtInput	**Mandatory**			
Object Name	Access	Type	Value	Description
DefaultID	Read-Write	Integer32		The ID of the default input device, i.e., the value of the corresponding input device's ID value.

Notes:

5.1. Input Table - Mandatory

prtInput	**Mandatory**		**prtInputTable**	
Object Name	Access	Type	Value	Description
ID	Read-Only	Integer32		A unique value used by the printer to identify this input device. Although these values may change due to a major reconfiguration of the device (e.g. the addition of new input devices to the printer), values are expected to remain stable across successive printer power cycles.
Type	Read-Only	Enum (2)	InputType	The type of technology (discriminated primarily according to feeder mechanism type) employed by the input device. Note, the Optional Input Class provides for a descriptor field to further qualify the other choice. This value is a type 2 enum.
DimUnit	Read-Write	Enum (1)	MediaUnit	The unit of measurement for use in calculating and relaying dimensional values for this input device. This value is a type 1 enum.
MediaDimXFeedDir	Read-Write	Integer32		This object provides the value of the current dimensions supported by the input device for measurements taken ninety degrees relative to the feed direction associated with that device in input device dimensional units (DimUnit). The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown.
MediaDimFeedDir	Read-Write	Integer32		The current dimensions supported by the input device for measurements taken parallel relative to the feed direction associated with that device in input device dimensional units (DimUnit). The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown.
CapacityUnit	Read-Write	Enum (1)	CapacityUnit	The unit of measurement for use in calculating and relaying capacity values for this input device. This value is a type 1 enum.
MaxCapacity	Read-Write	Integer32		The maximum capacity of the input sub-unit in input device capacity units (CapacityUnit). There is no convention associated with the media itself so this value reflects claimed capacity. This field is read-write in order to allow a management application to request the desired capacity input device. The value (-1) means other and specifically indicates that the device places no restrictions on this parameter. The value (-2) means unknown.
CurrentLevel	Read-Write	Integer32		The current capacity of the input sub-unit in input device capacity units (CapacityUnit). This field is read-write to accommodate remote op-panel implementation. The value (-1) means other and specifically indicates that the device places no restrictions on this parameter. The value (-2) means unknown.
Status	Read-Only	Enum (1)	EntStatus	The current status of this input device This value is a type 1 enum..
MediaName	Read-Write	String(63)		The name of the current medium contained in this input sub-uni. Examples: "iso-a4-white", "letter-white", "Hillary's Pink Presidential Letterhead". See Appendix C for standardized names that are recommended for use.

Notes:



5.2. Input Group Type Extended - Optional

prtInput - TypeExtended Optional			prtInputTableAugment	
Object Name	Access	Type	Value	Description
Name	Read-Write	String(63)		The name assigned to this input device.
VendorName	Read-Only	String(63)		The vendor name of this input device.
Model	Read-Only	String(63)		The model name of this input device.
Version	Read-Only	String(63)		The version of this input device.
SerialNumber	Read-Only	String(63)		The serial number assigned to this input device.
Description	Read-Only	String(255)		A free-form text description of this input device in the localization specified by **prtGeneralCurrentLocalization**.
Security	Read-Write	Enum (1)	YesNoOnOff	Indicates if this input device has some security associated with it.

Notes:

6. Media Group - Optional

The Media Extended Group supports identification of media installed or available for use on a printing device. Medium resources are identified by name, and include a collection of characteristic attributes that may further be used for selection and management of them. The Media Extended group is conceptually built in the form of an SNMP "AUGMENTS" construction off of the InputDevice table. In this manner, a minimally conforming implementation may choose to not support reporting of media resources if it cannot do so.

prtInput - MediaExtended	Optional		prtInputTableAugment	
Object Name	Access	Type	Value	Description
MediaWeight	Read-Write	Integer32		The weight of the medium in grams / per meter squared associated with this input device. The value (-1) means other and specifically indicates that the device places no restrictions on this parameter. The value (-2) means unknown.
MediaType	Read-Write	String(63)		The name of the type of medium associated with this input device using standardized string values from ISO 10175 (DPA) and ISO 10180 (SPDL) which are: **stationery** Separately cut sheets of an opaque material **transparency** Separately cut sheets of a transparent material **envelope** Envelopes that can be used for conventional mailing purposes **envelope-plain** Envelopes that are not preprinted and have no windows **envelope-window** Envelopes that have windows for addressing purposes **continuous-long** Continuously connected sheets of an opaque material connected along the long edge **continuous-short** Continuously connected sheets of an opaque material connected along the short edge **tab-stock** Media with tabs **multi-part-form** Form medium composed of multiple layers not pre-attached to one another; each sheet may be drawn separately from an input source **labels** Label stock **multi-layer** Form medium composed of multiple layers which are pre-attached to one another; e.g., for use with impact printers
MediaColor	Read-Write	String(63)		The name of the color of the medium associated with this input device using standardized string values from ISO 10175 (DPA) and ISO 10180 (SPDL) which are: **other** **unknown** **white** **pink** **yellow** **buff** **goldenrod** **blue** **green** **transparent** Implementors may add additional string values. The naming conventions in ISO 9070 are recommended in order to avoid potential name clashes.

MediaFormParts	Read-Write	Integer32		The number of parts associated with the medium associated with this input device if the medium is a multi-part form. The value (-1) means other and specifically indicates that the device places no restrictions on this parameter. The value (-2) means unknown.

Notes:

1. I added constraints of 63 characters for MediaType and MediaColor.
2. I added the standard string values to MediaType and MediaColor, since they aren't enums.
3. *Issue 34 - Should MediaType and MediaColor string value names be localized?*
4. *Issue 35 - Is a constraint of 63 octets enough for translating MediaType and MediaColor string values into all languages and character sets (two octet character sets use up two octets per character) sufficient?*

7. Output Group - Mandatory

Output sub-units are managed as a tabular, indexed collection of possible devices capable of receiving media delivered from the printing process. Output sub-units typically have a location, a type, an identifier, a set of constraints on possible media sizes and potentially other characteristics, and may be capable of indicating current status or capacity.

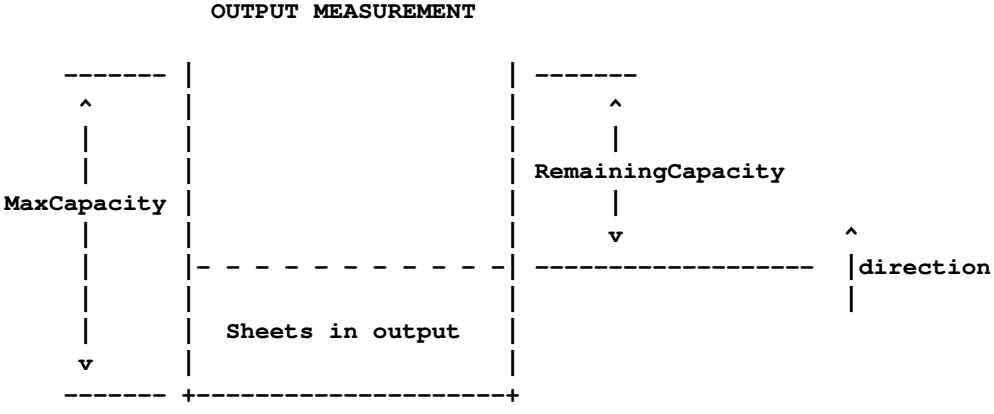
prtOutput	Mandatory			
Object Name	Access	Type	Value	Description
DefaultID	Read-Write	Integer32		The ID of the default output device, i.e., the value of the corresponding output device's ID value.

Notes:

7.1. Output Table - Mandatory

prtOutput	Mandatory		prtOutputTable	
Object Name	Access	Type	Value	Description
ID	Read-Only	Integer32		A unique value used by this printer to identify this output device. Although these values may change due to a major reconfiguration of the device (e.g. the addition of new output devices to the printer), values are expected to remain stable across successive printer power cycles.
Type	Read-Only	Enum (2)	OutputType	The type of technology supported by this output device. This value is a type 2 enum.
CapacityUnit	Read-Only	Enum (1)	CapacityUnit	The unit of measurement for use in calculating and relaying capacity values for this output device. This value is a type 1 enum.
MaxCapacity	Read-Write	Integer32		The maximum capacity of this output sub-unit in output device capacity units (CapacityUnit). There is no convention associated with the media itself so this value essentially reflects claimed capacity. This field is read-write in order to allow a management application to set the desired capacity of the output device. The value (-1) means other and specifically indicates that the device places no restrictions on this parameter. The value (-2) means unknown.
RemainingCapacity	Read-Write	Integer32		The remaining capacity of the possible output sub-unit capacity in output device capacity units (CapacityUnit) of this output device.
Status	Read-Only	Enum (1)	EntStatus	The current status of this output device. This value is a type 1 enum.

Notes:



7.2. Output Group - Type Extended Objects - Optional

prtOutput - TypeExtended	Optional		prtOutputTableAugment	
Object Name	Access	Type	Value	Description
Name	Read-Write	String(63)		The name assigned to this output device.
VendorName	Read-Only	String(63)		The vendor name of this output device.
Model	Read-Only	String(63)		The name assigned to this output device.
Version	Read-Only	String(63)		The version of this output device.
SerialNumber	Read-Only	String(63)		The serial number assigned to this output device.
Description	Read-Only	String(255)		A free-form text description of this output device in the localization specified by **prtGeneralCurrentLocalization**.
Security	Read-Write	Enum (1)	YesNoOnOff	Indicates if this output device has some security associated with it.

Notes:

7.3. Output - Dimensions Extended - Optional

prtOutput - DimExtended	Optional		prtOutputTableAugment	
Object Name	Access	Type	Value	Description
DimUnit	Read-Only	Enum (1)	MediaUnit	The unit of measurement for use in calculating and relaying dimensional values for this output device. This value is a type 1 enum.
MaxDimXFeedDir	Read-Write	Integer32		The maximum dimensions supported by this output device for measurements taken ninety degrees relative to the feed direction associated with that device in output device dimensional units (DimUnit).
MaxDimFeedDir	Read-Write	Integer32		The maximum dimensions supported by this output device for measurements taken parallel relative to the feed direction associated with that device in output device dimensional units (DimUnit).
MinDimXFeedDir	Read-Write	Integer32		The minimum dimensions supported by this output device for measurements taken ninety degrees relative to the feed direction associated with that device in output device dimensional units (DimUnit).
MinDimFeedDir	Read-Write	Integer32		The minimum dimensions supported by this output device for measurements taken parallel relative to the feed direction associated with that device in output device dimensional units (DimUnit).

Notes:

7.4. Output - Features - Optional

prtOutput - Features	Optional		prtOutputTableAugment	
Object Name	Access	Type	Value	Description
StackingOrder	Read-Write	Enum (1)	StackingOrder	The current state of the stacking order for the associated output device. “1 to n” means pages are output in the order in which they are acquired. “n to 1” means pages are output from last to first which implies that all pages must be acquired before they can be output. This value is a type 1 enum.
PageDeliveryOrientation	Read-Write	Enum (1)	PageDeliveryOrientation	The reading surface that will be "up" when pages are delivered to the associated output device. Values are Face-Up and Face-Down. (Note: interpretation of these values is in general context-dependent based on locale; presentation of these values to an end-user should be normalized to the expectations of the user). This value is a type 1 enum.
Bursting	Read-Write	Enum (1)	YesNoOnOff	By its presence, this object indicates that the output device supports bursting, and if so, whether the feature is enabled. Bursting is the process by which continuous media is separated into individual sheets, typically by bursting along pre-formed perforations. This value is a type 1 enum.

Decollating	Read-Write	Enum (1)	YesNoOnOff	By its presence, this object indicates that the output device supports decollating, and if so, whether the feature is enabled. Decollating is the process by which the individual parts within a multi-part form are separated and sorted into separate stacks for each part. This value is a type 1 enum.
PageCollated	Read-Write	Enum (1)	YesNoOnOff	By its presence, this object indicates that the output device supports page collation, and if so, whether the feature is enabled. This value is a type 1 enum.
JobCollated	Read-Write	Enum (1)	YesNoOnOff	By its presence, this object indicates that the output device supports job collation, and if so, whether the feature is enabled. This value is a type 1 enum.
OffsetStacking	Read-Write	Enum (1)	YesNoOnOff	By its presence, this object indicates that the output device supports offset stacking, and if so, whether the feature is enabled. This value is a type 1 enum.
Slipsheet	Read-Write	Enum (1)	YesNoOnOff	By its presence, this object indicates that the output device supports slipsheet insertion between sets, and if so, whether the feature is enabled. This value is a type 1 enum.

Notes:

8. Finisher Group - Optional

The Finisher Group is an optional group, and is included in this specification for completeness due to its relationship to the Output Group.

In general, finishers are very complicated devices and their management is a topic suitable for considerable, additional investigation. The goal of this class is therefore not to standardize management of arbitrarily complex finishers. Rather, this class only standardizes basic management objects suitable for a very generic abstract finisher. The value of this, is that by defining this basic class we establish the architectural framework for a common approach which can then be augmented as required in a consistent manner by vendors and consortia.

Finisher devices are typically complicated systems and can (or should be) dealt with largely isolated from the "output device." Since there are so many configuration options for finisher devices (and some of them may indeed be integrated within the output device itself), the question is whether to deal with finishers as part of output device management, or whether to deal with them separately. The merit of separating the two "types" of devices is that it enables greater modularity across the widest variety of interesting configurations, and that is the approach taken in this specification.

8.1. Output Associated Finishers Table - Optional

prtOutput - Associated Finisher Optional			**prtOutputAssocFinisher**	
Object Name	Access	Type	Value	Description
ID	Read-Only	Integer32		The ID of the output device to which the finisher is connected.
Stitching	Read-Only	Enum (1)	YesNoOnOff	By its presence, this object indicates that the output device supports stapling as a process accessible in the production path associated with it. The value returned by this object is the identifier for the associated finishing device; the value serves as the index for identifying the device in the **prtFinisherTable**. This value is a type 1 enum.
Folding	Read-Only	Enum (1)	YesNoOnOff	By its presence, this object indicates that the output device supports folding as a process accessible in the production path associated with it. The value returned by this object is the identifier for the associated finishing device; the value serves as the index for identifying the device in the **prtFinisherTable**. This value is a type 1 enum.
Binding	Read-Only	Enum (1)	YesNoOnOff	By its presence, this object indicates that the output device supports binding as a process accessible in the production path associated with it. The value returned by this object is the identifier for the associated finishing device; the value serves as the index for identifying the device in the **prtFinisherTable**. This value is a type 1 enum.
Punching	Read-Only	Enum (1)	YesNoOnOff	By its presence, this object indicates that the output device supports punching as a process accessible in the production path associated with it. The value returned by this object is the identifier for the associated finishing device; the value serves as the index for identifying the device in the **prtFinisherTable**. This value is a type 1 enum.
Trimming	Read-Only	Enum (1)	YesNoOnOff	By its presence, this object indicates that the output device supports trimming as a process accessible in the production path associated with it. The value returned by this object is the identifier for the associated finishing device; the value serves as the index for identifying the device in the **prtFinisherTable**. This value is a type 1 enum.
Insertion	Read-Only	Enum (1)	YesNoOnOff	By its presence, this object indicates that the output device supports inserting as a process accessible in the production path associated with it. The value returned by this object is the identifier for the associated finishing device; the value serves as the index for identifying the device in the **prtFinisherTable**. This value is a type 1 enum.

Notes:

8.2. Finishing Configuration - Optional

The finishing table maintains mappings between configurations of possibly zero or more finishers and zero or more output devices. Each output device may provide a value for the optional column "**prtOutputAssociatedFinisher**". The value in that column is an integer which is the "**prtFinisherConfigurationId**". The value for the "**prtFinisherConfigurationId**" is the index into the "**prtFinisherTable**".

prtFinisher	Optional			
Object Name	Access	Type	Value	Description
DefaultID	Read-Write	Integer32		The ID of the default finisher device, i.e., the value of the corresponding finisher device's **ID** value. A value of 0 means that a finisher is not used in default configurations.

Notes:

8.3. Finisher Table - Optional

prtFinisher	Optional		prtFinisherTable	
Object Name	Access	Type	Value	Description
ID	Read-Only	Integer32		A unique value used by the printer to identify this finisher device. Although these values may change due to a major reconfiguration of the device (e.g. the addition of new finisher devices to the printer), values are expected to remain stable across successive printer power cycles.
Type	Read-Only	Enum (2)	FinisherType	The type of finishing performed by this finisher device. This value is a type 2 enum.
NamedFinishing	Read-Only	Enum (2)	NamedFinishing	The finishing operations available through this finishing device. This value is a type 2 enum.
ConsumableUnit	Read-Only	Enum (1)	FinisherConsumableUnit	The consumable resource unit appropriate to this finisher device and its operation. This value is a type 1 enum.
MaxConsumableCapacity	Read-Write	Integer32		The maximum capacity of this finisher in consumable resource units of measure (**ConsumableUnit**). This value is particularly suited for providing a gauge that can reflect the maximum consumable resource capacity of this finisher.
CurrentConsumableCapacity	Read-Write	Integer32		The current capacity of this finisher in consumable resource units of measure (**ConsumableUnit**). This value is particularly suited for providing a gauge that can reflect the current available consumable resource capacity of this finisher.
Status	Read-Only	Enum (1)	EntStatus	The current status of this finisher. This value is a type 1 enum.

Notes:

9. Marker Group - Mandatory

A marker is the mechanisms that produces marks on the print media. The marker sub-units and their associated supplies are represented by the Marker Group in the model. A printer can contain one or more marking mechanisms. Some examples of multiple marker sub-units are: a printer with separate markers for normal and magnetic ink or an imagesetter that can output to both a proofing device and final film. Each marking device can have its own set of characteristics associated with it, such as marking technology and resolution.

prtMarker	**Mandatory**			
Object Name	Access	Type	Value	Description
DefaultID	Read-Write	Integer32		The ID of the default marker device, i.e., the value of the corresponding marker device's ID value.

Notes:

9.1. Marker Table - Mandatory

The printable area margins as listed below define an area of the print media which is guaranteed to be printable for all combinations of input, media paths, and interpreters for this marker.

prtMarker	**Mandatory**	**prtMarkerTable**		
Object Name	Access	Type	Value	Description
ID	Read-Only	Integer32		A unique value used by the printer to identify this marking device. Although these values may change due to a major reconfiguration of the device (e.g. the addition of new marking devices to the printer), values are expected to remain stable across successive printer power cycles.
MarkTech	Read-Only	Enum (2)	MarkingTech	The type of marking technology used for this marking device. This value is a type 2 enum.
CounterUnit	Read-Only	Enum (1)	CounterUnit	The unit that will be used by the printer when reporting counter values for this marking device. The time units of measure are provided for a device like a strip recorder that does not or cannot track the physical dimensions of the media and does not use characters, lines or sheets. This value is a type 1 enum.
LifeCount	Read-Only	Counter32		The count of the number of units of measure counted during the life of printer using units of measure as specified by CounterUnit.
PowerOnCount	Read-Only	Counter32		The count of the number of units of measure counted since the equipment was most recently powered on using units of measure as specified by CounterUnit.
ProcessColorants	Read-Only	Integer32		The number of process colors supported by this marker. A process color of 1 implies monochrome. The value of this object and SpotColorants cannot both be 0. A value of (-1) implies 'other' or 'infinite' while a value of (-2) implies 'unknown'.
SpotColorants	Read-Only	Integer32		The number of spot colors supported by this marker. The value of this object and ProcessColorants cannot both be 0. A value of (-1) implies 'other' or 'infinite' while a value of (-2) implies 'unknown'
AddressabilityUnit	Read-Only	Enum (1)	AddrUnit	The unit of measure of the denominator of Horizontal and Vertical addressability. This value is a type 1 enum.
AddressabilityFeedDir	Read-Only	Integer32		The addressability of the engine in the feed direction in units of measure specified by AddressabilityUnit. A value of (-1) implies 'other' or 'infinite' while a value of (-2) implies 'unknown'.
AddressabilityXFeedDir	Read-Only	Integer32		The addressability of this marker in the cross feed direction in units of measure specified by AddressabilityUnit. A value of (-1) implies 'other' or 'infinite' while a value of (-2) implies 'unknown'.
NorthMargin	Read-Only	Integer32		The margin from the leading edge of the medium in units identified by AddressabilityUnit. Printing within this margin (and all others) is guaranteed for all interpreters. The value (-2) means unknown.

SouthMargin	Read-Only	Integer32		The margin from the trailing edge of the medium in units identified by AddressabilityUnit. Printing within this margin (and all others) is guaranteed for all interpreters. The value (-2) means unknown.
WestMargin	Read-Only	Integer32		The margin from the left edge of the medium in units identified by AddressabilityUnit. Printing within this margin (and all others) is guaranteed for all interpreters. The value (-2) means unknown.
EastMargin	Read-Only	Integer32		The margin from the right edge of the medium in units identified by AddressabilityUnit. Printing within this margin (and all others) is guaranteed for all interpreters. The value (-2) means unknown.
Status	Read-Only	Enum (1)	EntStatus	The current status of this marker device. This value is a type 1 enum.

Notes:

9.2. Marker Supplies Table - Optional

prtMarkerSupplies	**Mandatory**			**prtMarkerSuppliesTable**
Object Name	Access	Type	Value	Description
ID	Read-Only	Integer32		A unique value used by the printer to identify this marker supply. Although these values may change due to a major reconfiguration of the device (e.g. the addition of new marker supplies to the printer), values are expected to remain stable across successive printer power cycles.
MarkerID	Read-Only	Integer32		The Marker ID of the marking device with which this marker supply entity is associated.
ColorantID	Read-Only	Integer32		The Colorant ID of the colorant with which this marker supply entity is associated. This value shall be 0 if there is no colorant table.
Class	Read-Only	Enum (1)	MkrSupClass	Indicates whether this supply entity represents a supply container that is consumed or a receptacle that is filled. This value is a type 1 enum.
Type	Read-Only	Enum (3)	MkrSupType	The type of this supply. This value is a type 3 enum.
Description	Read-Only	String(255)		The description of this supply container/receptacle in the localization specified by **prtGeneralCurrentLocalization**.
SupplyUnit	Read-Only	Enum (1)	MkrSupUnit	Unit of this marker supply container/receptacle. This value is a type 1 enum.
MaxCapacity	Read-Only	Integer32		The maximum capacity of this supply container/receptacle expressed in SupplyUnit.
Level	Read-Only	Integer32		The current level if this supply is a container; the remaining space if this supply is a receptacle.

Notes

1. Deleted the redundant **MarkerSupply** from **prtMarkerSuppliesMarkerSupplyID** to give: **prtMarkerSuppliesID**

9.3. Marker Colorant Table - Optional

prtMarkerColorant	**Optional**			**prtMarkerColorantTable**
Object Name	Access	Type	Value	Description
ID	Read-Only	Integer32		A unique value used by the printer to identify this colorant. Although these values may change due to a major reconfiguration of the device (e.g. the addition of new colorants to the printer), values are expected to remain stable across successive printer power cycles.
MarkerID	Read-Only	Integer32		The marker ID with which this colorant entry is associated.
Role	Read-Only	Enum (1)	ColorantRole	The role played by this colorant. This value is a type 1 enum.

Value	Read-Only	String(63)		The name of the color of this colorant using standardized string names from ISO 10175 (DPA) and ISO 10180 (SPDL) which are: **other** **unknown** **white** **red** **green** **blue** **cyan** **magenta** **yellow** **black** Implementors may add additional string values. The naming conventions in ISO 9070 are recommended in order to avoid potential name clashes
Tonality	Read-Only	Integer32		The distinct levels of tonality realizable by a marking device when using this colorant. This value does not include the number of levels of tonal difference that an interpreter can obtain by techniques such as half toning. This value must be at least 2.

Notes:

1. I copied in the standard colorant value strings.

2. Deleted the redundant **Colorant** from **prtMarkerColorantColorantID** to give: **prtMarkerColorantID**

10. Media Path Group - Mandatory

The media paths encompass the mechanisms in the printer that move the media through the printer and connect all other media related sub-units: inputs, outputs, markers and finishers. A printer contains one or more media paths. These are represented by the Media Path Group in the model. The Media Path group has some attributes that apply to all paths plus a table of the separate media paths.

prtMediaPath	**Mandatory**			
Object Name	Access	Type	Value	Description
DefaultID	Read-Write	Integer32		The ID of the default media path, i.e., the value of the corresponding media path's ID value.
MaxSpeedPrintUnit	Read-Only	Enum (1)	SpeedPrintUnit	The unit of measure used in specifying the speed of all media paths in the printer, i.e., the numerator of the units of measure of the media path speed. This value is a type 1 enum.
MaxSpeedTimeUnitMaxSpeed TimeUnit	Read-Only	Enum (1)	SpeedTimeUnit	The unit of time used in specifying the speed of all media paths in the printer, i.e., the denominator of the units of measure of the media path speed. This value is a type 1 enum.
MediaSizeUnitMediaSizeUnit	Read-Only	Enum (1)	MediaUnit	The units of measure of media size for use in calculating and relaying dimensional values for all media paths in the printer. This value is a type 1 enum.

Notes:

10.1. Media Path Table - Optional

prtMediaPath	**Mandatory**		**prtMediaPathTable**	
Object Name	Access	Type	Value	Description
ID	Read-Only	Integer32		A unique value used by the printer to identify this media path. Although these values may change due to a major reconfiguration of the device (e.g. the addition of new media paths to the printer), values are expected to remain stable across successive printer power cycles.
MaxSpeed	Read-Only	Integer32		The maximum printing speed of this media path expressed in media path speed units (units of MaxSpeedPrintUnit per MaxSpeedTimeUnit). A value of (-1) implies 'other', while a value of (-2) implies 'unknown'.
MaxMediaFeedDir	Read-Only	Integer32		The maximum physical media size in the feed direction of this media path expressed in units of measure specified by MediaSizeUnit. A value of (-1) implies 'other' or 'infinite' while a value of -(-2) implies 'unknown'. A value of 0 implies unlimited.
MaxMediaXFeedDir	Read-Only	Integer32		The maximum physical media size across the feed direction of this media path expressed in units of measure specified by MediaSizeUnit. A value of (-1) implies 'other' or 'infinite' while a value of (-2) implies 'unknown'.
MinMediaFeedDir	Read-Only	Integer32		The minimum physical media size in the feed direction of this media path expressed in units of measure specified by MediaSizeUnit. A value of (-1) implies 'other' while a value of (-2) implies 'unknown'.
MinMediaXFeedDir	Read-Only	Integer32		The minimum physical media size across the feed direction of this media path expressed in units of measure specified by MediaSizeUnit. A value of (-2) implies 'unknown'.
Type	Read-Only	Enum (2)	MediaPathType	The type of media path for this media path. This value is a type 2 enum.
Description	Read-Only	String(255)		The manufacturer-provided description of this media path in the localization specified by **prtGeneralCurrentLocalization**.
Status	Read-Only	Enum (1)	EntStatus	The current status of this media path. This value is a type 1 enum.

Notes:

1. Changed **Across** to **X** to agree with Interpreter group.
2. Shortened prtMediaPathMediaPathType to prtMediaPathType and prtMediaPathMediaPathDescription to prtMediaPathDescription.

11. Channel Group - Mandatory

Channels are sources of print data. There are many kinds of channels; some of which are based on networks and others which are not. For example, a channel can be a serial (or parallel) connection; it can be a service, such as the Unix Line Printer Daemon (LPD), offering itself over a network connection (interface); or it could be a disk drive into which a floppy disks with the print data is inserted.

Channel example	Implementation
serial port channel	bi-directional data channel
parallel port channel	often uni-directional channel
IEEE 1284 port channel	bi-directional channel
SCSI port channel	bi-directional
Apple PAP channel	may be based on Local-, Ether- or TokenTalk
LPD Server channel	typically TCP/IP based, port 515
Novell Remote Printer channel	typically SPX/IPX based
Novell Print Server channel	typically SPX/IPX based
port 9100 channel	HP and friends
Adobe AppSocket(9101) channel	a bi-directional extension of LPD

It is easy to note that this is a mixed bag. There are some physical connections over which no (or meager) protocols are run (e.g. the serial or old parallel ports) and there are services which often have elaborate protocols that run over a number of protocol stacks. In the end what is important is the delivery of print data through the channel.

Using the OSI seven layer model of for data interchange, the "channel" fits most accurately into the Application layer. Although this is not entirely clean (especially for the serial and old-parallel ports) it is clear that the interpreters that are consuming the print data are running at an application level.

11.1. The channel table and its underlying structure

The first five items in the Channel Table define the "channel" itself as seen at the OSI Application Layer (layer 7). This channel depends on lower layers of the OSI protocol, most often layers 1-4, to provide the data that flows through the channel. (Layer 1 is the physical connection; layer 2 is the data-link protocol, such as EtherNet; layer 3 is the inter-networking protocol, such as IP; and layer 4 is the transport protocol, such as TCP.) (Layers 5 (session) and 6(presentation) are most often irrelevant to the channel, but layer 6 could be used for character code conversion and/or decompression. A printer may be connected to print data sources in more than one way which means that there may be multiple variants at each of layers 1-4. Hence some way is needed to specify the "underlying" structure of each channel.

What is this underlying structure? A serial channel, has very little structure, except perhaps a protocol to allow binary data to be sent transparently over that channel along with control information. A network connection will typically have one or more variants at each of levels 1-4 of the OSI model. An example channel might have the Novell Remote Printer application protocol running on top of SPX (at layer 4) running on top of IPX (at layer 3) running on top of EtherNet (at layer 2) running over a large EtherNet cable (at layer 1).

In SNMP, there are groups which represent these various layers: the Interface Groups represent layers 1 and 2, including sub-layers. The inter-networking groups, such as the IP group represent layer 3 and its associated addressing. The Transmission groups, such as the TCP group and its associated connection table represent layer 4. The ifTable in the Interfaces Group has a which represents the various possible interfaces up to the Interment Level (level 3). There are no corresponding structures within SNMP that represent higher levels. For this reason, for this standard, the underlying structure will be represented only by identifying the topmost interface that underlies each channel.

11.2. Channel Table - Mandatory

The **prtChannelTable** represents the set of input data sources which can provide print data to one or more of the interpreters available on a printer

prtChannel	Mandatory		prtChannelTable	
Object Name	Access	Type	Value	Description

ID	Read-Only	Integer32		A unique value used by the printer to identify this data channel. Although these values may change due to a major reconfiguration of the device (e.g. the addition of new data channels to the printer), values are expected to remain stable across successive printer power cycles.
CurrentJobCntlLangID	Read-Write	Integer32		The value of this attribute is an Index that selects an entry in the **prtInterpreterTable**. This interpreter defines the syntax used for control functions, such as querying or changing environment variables and identifying job boundaries (e.g. PJL, PostScript, NPAP).
DefaultPageDescLangID	Read-Write	Integer32		The value of this attribute is an Index that selects an entry in the **prtInterpreterTable**. This interpreter defines the default Page Description Language interpreter to be used for the print data unless the Control Language is used to select a specific interpreter (e.g., PCL, PostScript Language, auto-sense).
State	Read-Write	Enum (1)	ChannelState	The state of this print data channel. The value determines whether control information and print data is allowed through this channel or not. This value is a type 1 enum.
Type	Read-Only	Enum (2)	ChannelType	The type of this print data channel. This value is a type 2 enum.
ifIndex	Read-Write	Integer32		The value of ifIndex (in the ifTable; see the interface section of MIB-2/RFC 1573) which corresponds to this channel. When more than one row of the ifTable is relevant, this is the index of the row representing the topmost layer in the if hierarchy.
Status	Read-Only	Enum (1)	EntStatus	The current status of the channel. This value is a type 1 enum.

Notes:

1. Removed and/or in State, since Issue 27 says control and print data aren't controlled separately.

12. Interpreter Group - Mandatory

The interpreter sub-units are responsible for the conversion of a description of intended print instances into images that are to be marked on the media. A printer may have one or more interpreters. The interpreter sub-units are represented by the Interpreter Group in the Model. Each interpreter is generally implemented with software running on the System Controller sub-unit. The Interpreter Table has one entry per interpreter where the interpreters include both Page Description Language (PDL) Interpreters and Control Language Interpreters.

prtInterpreter	**Mandatory**			
Object Name	Access	Type	Value	Description
DefaultID	Read-Write	Integer32	IntpLangFam	The ID of the default interpreter, i.e., the value of the corresponding interpreter's ID, to be used in case no device specific method is used to specify which interpreter should be used. Priority on selecting the interpreter are 1) Commands in the data stream; 2) The default interpreter for the channel on which the data arrives; and lastly 3) This object.

12.1. Interpreter Table - Mandatory

The **prtInterpreterTable** is a table representing the interpreters in the printer. An entry shall be placed in the interpreter table for each interpreter on the printer.

prtInterpreter	**Mandatory**		**prtInterpreterTable**	
Object Name	Access	Type	Value	Description
ID	Read-Only	Integer32		A unique value for each PDL or control language for which there exists an interpreter or emulator in the printer. The value is used to identify this interpreter. Although these values may change due to a major reconfiguration of the device (e.g. the addition of new interpreters to the printer), values are expected to remain stable across successive printer power cycles.
LangFamily	Read-Only	Enum (2)	IntpLangFam	The family name of a Page Description Language (PDL) or control language which this interpreter in the printer can interpret or emulate. This type 2 list of enumerations requires review before additional entries are made. This value is a type 2 enum.
LangLevel	Read-Only	String(31)		The level of the language which this interpreter is interpreting or emulating. This might contain a value like '5e' for an interpreter which is emulating level 5e of the PCL language. It might contain '2' for an interpreter which is emulating level 2 of the PostScript language. Similarly it might contain '2' for an interpreter which is emulating level 2 of the HPGL language.
LangVersion	Read-Only	String(31)		The date code or version of the language which this interpreter is interpreting or emulating.
Description	Read-Only	String(255)		A string to identify this interpreter in the localization specified by **prtGeneralCurrentLocalization** as opposed to the language which is being interpreted. It is anticipated that this string will allow manufacturers to unambiguously identify their interpreters.
Version	Read-Only	String(31)		The date code, version number, or other product specific information tied to this interpreter. This value is associated with the interpreter, rather than with the version of the language which is being interpreted or emulated.
FeedAddressability	Read-Only	Integer32		The maximum interpreter resolution in the feed direction in **prtMarkerAddressabilityUnit** (the same units of measure used by the object used to specify marker addressability) for this interpreter. If this interpreter has no limit on addressability the value is zero. The value (-1) means other and specifically indicates that the device places no restrictions on this parameter. The value (-2) means unknown.

XFeedAddressability	Read-Only	Integer32		The maximum interpreter resolution in the across feed direction in **prtMarkerAddressabilityUnit** (the same units of measure used by the object used to specify marker addressability) for this interpreter. If this interpreter has no limit on addressability the value of this object is zero. The value (-1) means other and specifically indicates that the device places no restrictions on this parameter. The value (-2) means unknown.
DefaultCharSetIn	Read-Write	Integer32		An index into the CharacterSetTable that identifies the default character set for data into this interpreter.
DefaultCharSetOut	Read-Write	Integer32		An index into the CharacterSetTable that identifies the default character set for data coming from this interpreter through the printer's output channel
TwoWay	Read-Only	Enum (1)	YesNoOnOff	Indicates whether or not this interpreter returns information back across the data channel to the host. This value is a type 1 enum.

Notes:

13. Console Group - Mandatory

Many printers have a console on the printer, the operator console, that is used to display and modify the state of the printer. The console can be as simple as a few indicators and switches or as complicated as full screen displays and keyboards. There can be at most one such console.

prtConsole Mandatory				
Object Name	Access	Type	Value	Description
Localization	Read-Write	Integer32		An index into localization table giving the language, country, and character set to be used for the console.
NumberOfDisplayLines	Read-Only	Integer32		The number of lines on printers physical display.
NumberOfDisplayChars	Read-Only	Integer32		The number of characters per line displayed on the physical display.
Lock	Read-Write	Enum (2)	LockState	A lock to provide one-at-a-time access by co-operating management applications to the operation of the operators console. This value is a type 2 enum.

Notes:

1, The prefix for all objects in the Console table is **prtConsole**, so the rest of the name should not repeat the word Console.

2. I deleted the word Operator. Its too confusing to sometimes use Operator Console and sometimes Console.

13.1. Display Buffer Table - Mandatory

prtConsole Mandatory prtDisplayBufferTable				
Object Name	Access	Type	Value	Description
LineText	Read-Write	String(255)		The content of a line in the logical display buffer of the operator's console of the printer. When a write operation occurs, normally a critical message, to one of the LineText strings, the agent should make that line displayable if a physical display is present. The printer should not overwrite this line once written until a zero length string is written to unlock that line. Printer generated strings should be in the localization specified by ConsoleLocalization. Management Application generated strings should be localized by the Management Application.

Notes:

prtConsole Mandatory prtConsoleLightsTable				
Object Name	Access	Type	Value	Description
LightID	Read-Only	Integer32		A unique value used by the printer to identify this light. Although these values may change due to a major reconfiguration of the device (e.g. the addition of new lights to the printer), values are expected to remain stable across successive printer power cycles.
OnTime	Read-Only	Integer32		The on time in milliseconds of blinking of this light; 0 indicates off always.
OffTime	Read-Only	Integer32		The off time in milliseconds of blinking of this light; 0 indicates on always.
Color	Read-Only	Enum (2)	LightColor	The color of this light. This value is a type 2 enum.
Description	Read-Only	String(255)		The vendor description or label of this light in the localization specified by ConsoleLocalization.

Notes:

14. Alerts Group - Mandatory

The **prtAlertTable** lists all the critical and non-critical alerts currently active in the printer. A critical alert is one that stops the printer from printing immediately and printing can not continue until the critical alert condition is eliminated. Non-critical alerts are those items that do not stop printing but may at some future time. The table contains information on the severity, component, detail location within the component, alert code and description of each critical alert that is currently active within the printer. See 2.2.13 for a more complete description of the alerts table and its management.

prtAlert	**Mandatory**		**prtAlertTable**	
Object Name	Access	Type	Value	Description
Index	Read-Only	Integer32		The index value used to determine which alerts have been added or removed from the alert table. This in an incrementing integer starting from zero every time the printer is powered on. When the printer adds an alert to the table, that alert is assigned the next higher integer value from the last item entered into the table. If the index value reaches its maximum value, the next item entered will cause the index value to roll over and start at zero again. The first event placed in the alert table after power up initialization of the printer shall have an index value of 1. NOTE: The management application will read the alert table when a trap or event notification occurs or at a periodic rate and then parse the table to determine if any new entries were added by comparing the last know index value with the current highest index value. The management application will then update its copy of the alert table. When the printer discovers that an alert is no longer active, the printer shall remove the row for that alert from the table and shall reduce the number of rows in the table. The printer may add or delete any number of rows from the table at any time.
SeverityLevel	Read-Only	Enum (1)	SeverityLevelAlertCode	The level of severity of this alert table entry. The printer determines the severity level assigned to each entry into the table. This value is a type 1 enum.

TrainingLevel	Read-Only	Enum (2)	TrainingLevel	The level of training required to handle this alert. The training level is an enumeration that is determined and assigned by the printer manufacturer based on the information or the training required to handle this alert. The printer will break alerts into these different training levels. It is the responsibility of the management application in the system to determine how a particular alert is handled and how and to whom that alert is routed. The following are the four training levels of alerts: Field Service - Alerts that typically require advanced training and technical knowledge of the printer and its sub-units. An example of a technical person would be a manufacture's Field Service representative, or other person formally trained by the manufacturer or similar representative. Trained - Alerts that require an intermediate or moderate level of knowledge of the printer and its sub-units. A typical examples of alerts that a trained operator can handle is replacing toner cartridges. Untrained - Alerts that can be fixed without prior training either because the action to correct the alert is obvious or the printer can help the untrained person fix the problem. A typical example of such an alert is reloading paper trays and emptying output bins on a low end printer. Management - Alerts that have to do with overall operation of and configuration of the printer. Examples of management events are configuration change of sub-units. This value is a type 2 enum.
Group	Read-Only	Enum (1)	AlertGroupCode	The type of sub-unit within the printer model that this alert is related. Input, output, and markers are examples of printer model groups, i.e., examples of types of sub-units. This value is a type 1 enum.
GroupIndex	Read-Only	Integer32		An integer designation, assigned by the printer manufacturer, of the specific sub-unit within the group that caused this alert. The combination of the Group and the GroupIndex defines exactly which printer sub-unit caused the alert. Input #3, Output #2, and Marker #1 are examples.
Location	Read-Only	Integer32		The sub-unit location that is defined by the printer manufacturer to further refine the location of this alert with the designated sub-unit. The location is used in conjunction with the Group and GroupIndex values. Example: There is an alert in Input #2 at location number 7.
AlertCode	Read-Only	Enum (2)	*Xxxx*AlertCode	The code that describes the type of alert for this entry in the table. There are different codes for each sub-unit type. Jam, Media Low, Toner Out, and Output Full are examples of different codes for various sub-unit types. This value is a type 2 enum.
Description	Read-Only	String(255)		A description of this alert entry in the localization specified by **prtGeneralCurrentLocalization**. The description is provided by the printer to further elaborate on the enumerated alert or provide information in the case where the code is classified as “other” or “unknown”. The printer is required to return a description string but the string may be a null string.

Notes:

15. Enumerated Values

This section lists the enumerated values that have been assigned.

This working group has defined several type of enumerations. These enumerations differ in the method employed to control the addition of new enumerations. Throughout this document, references to "enum (n)", where n can be 1, 2 or 3 can be found in the various tables. The definitions of these types of enumerations are:

enum (1) All the values are defined in the Printer MIB/MIF specification (RFC for the Printer MIB). Additional enumerated values require a new RFC.

enum (2) An initial set of values are defined in the Printer MIB/MIF specification. Additional enumerated values are registered after review by this working group. The initial versions of the MIB and MIF forms of this specification will contain the values registered so far. After the MIB and MIF forms of this specification are approved, additional values will be registered through IANA *after* approval by this working group.

enum (3) An initial set of values are defined in the Printer MIB/MIF specification. Additional enumerated values are registered without working group review. The initial versions of the MIB and MIF forms of this specification will contain the values registered so far. After the MIB and MIF forms of this specification are approved, additional values will be registered through IANA *without* approval by this working group.

15.1. Sub-Unit Status

The sub-unit status values indicate the status of the sub-unit.

EntStatus (enum type 1) (used > 1)	
1	Other
2	Unknown
3	Ready Available
4	Critical
5	Noncritical
6	Both Critical And Noncritical
7	On Request
8	Unavailable
9	Busy Temporally Unavailable

15.2. Units of measure

The units of measure enumerated values indicate the unit of measure in which the values of other objects/attributes are currently being interpreted. The values of the different enumerations have been harmonized so that the same numeric value is used for the same unit in all enumerations.

AddrUnit (enum type 1)		MediaUnit (enum type 1) (used > 1)		CounterUnit (enum type 1)	
1	Other	1	Other	1	Other
2	Unknown	2	Unknown	2	Unknown
3	Thousandths Of Inches	3	Thousandths Of Inches	3	Thousandths Of Inches
4	Micrometers	4	Micrometers	4	Micrometers
		8	Sheets	5	Characters
				6	Lines
				7	Impressions
				8	Sheets
				9	Dot Row
				11	Hours
				16	Feet
				17	Meters
SpeedPrintUnit (enum type 1)		SpeedTimeUnit (enum type 1)		FinisherConsumableUnit (enum type 1)	
1	Other	1	Other	1	Other
2	Unknown	2	Unknown	2	Unknown
3	Thousandths Of inches	11	Hours	3	Thousandths Of Inches
4	Micrometers			4	Micrometers
5	Characters			8	Sheets
6	Lines			12	Thousandths Of Ounces
7	Impressions			13	Tenths Of grams
8	Sheets			14	Hundredths Of fluid ounces
9	Dot Row			15	Tenths Of milliliters
				16	Feet
				17	Meters
				18	Each
MkrSupplyUnit (enum type 1)		CapacityUnit (enum type 1) (used >1)			
1	Other	1	Other		
2	Unknown	2	Unknown		
3	Thousandths Of Inches	3	Thousandths Of Inches		
4	Micrometers	4	Micrometers		
8	Sheets	8	Sheets		
12	Thousandths Of Ounces	16	Feet		
13	Tenths Of Grams	17	Meters		
14	Hundredths Of Fluid Ounces				
15	Tenths Of Milliliters				

15.3. Device Types

InputType (enum type 2)		OutputType (enum type 2)		MediaPathType (enum type 2)	
1	Other	1	Other	1	Other
2	Unknown	2	Unknown	2	Unknown
3	Sheet Feed Auto Removable Tray	3	Removable Bin	3	Long Edge Binding Duplex
4	Sheet Feed Auto Non Removable Tray	4	Unremovable Bin	4	Short Edge Binding Duplex
		5	Continuous Roll Device	5	Either Edge Duplex
5	Sheet Feed Manual	6	Mail Box	6	Simplex
6	Continuous Roll				
7	Continuous Fan Fold				

FinisherType (enum type 2) 1 Other 2 Unknown 3 Stitching 4 Binding 5 Trimming 6 Die Cutting 7 Punching 8 Perforating 9 Slitting 10 Inserting 11 Covers 12 Folding		MkrSupplyClass (enum type 1) 1 Other 2 Unknown 3 Supply That Is Consumed 4 Receptacle That Is Filled
MarkingTechnology (enum type 2) 1 Other 2 Unknown 3 Electrophotographic LED 4 Electrophotographic Laser 5 Electrophotographic Other 6 Impact Moving Head Dot Matrix 9pin 7 Impact Moving Head Dot Matrix 24pin 8 Impact Moving Head Dot Matrix Other 9 Impact Moving Head Fully Formed 10 Impact Band 11 Impact Other 12 Inkjet Aqueous 13 Inkjet Solid 14 Inkjet Other 15 Pen 16 Thermal Transfer 17 Thermal Sensitive 18 Thermal Diffusion 19 Thermal Other 20 Electroerosion 21 Electrostatic 22 Photographic Microfiche 23 Photographic Imagesetter 24 Photographic Other 25 Ion Deposition 26 E beam 27 Typesetter		MkrSupplyType (enum type 3) 1 Other 2 Unknown 3 Toner 4 Waste Toner 5 Ink 6 Ink Cartridge 7 Ink Ribbon 8 Waste Ink 9 OPC 10 Developer 11 Fuser Oil 12 Solid Wax 13 Ribbon Wax 14 Waste Wax

15.4. Miscellaneous

MediaOrientation (enum type 1) 1 Other 2 Unknown 3 Portrait 4 Reverse Portrait 5 Landscape 6 Reverse Landscape		LineStatus 1 Other 2 Unknown 3 On Line 4 Pending On Line 5 Off Line 6 Pending Off Line
NamedFinshing (enum type 2) 1 Other 2 Unknown 3 Staple 4 Staple Top Left 5 Staple Bottom Left 6 Stable Top Right 7 Staple Bottom Right 8 Saddle Stitch 9 Edge Stitch 10 Punch 11 Cover 12 Bind		LightColor (enum type 2) 1 Other 2 Unknown 3 White 4 Red 5 Green 6 Blue 7 Cyan 8 Magenta 9 Yellow
YesNoOnOff (enum type 1) (used > 1) 1 Other 2 Unknown 3 Yes On 4 No Off	ResetType (enum type 3) 3 Power Cycle Reset 4 Reset NVRAM 5 Reset To Factory Defaults	ColorantRole (enum type 1) 1 Other 2 Unknown 3 Process 4 Spot
TrainingLevel (enum type 2) 1 Other 2 Unknown 3 Untrained 4 Trained 5 Field Service 6 Management	ChannelState (enum type 1) 1 Other 2 Unknown 3 Print And Control Data Accepted 4 No Data Accepted	PageDeliveryOrientation (enum type 1) 1 Other 2 Unknown 3 Face Up 4 Face Down
StackingOrder (enum type 1) 1 Other 2 Unknown 3 One To N 4 N To One	LockState (enum type 2) 1 Other 2 Unknown 3 Locked 4 Unlocked	SkinStatus (enum type 2) 1 Other 2 Unknown 3 Door Open 4 Door Closed 5 Interlock Open 6 Interlock Closed

Notes

1. MediaColor is supposed to be string values, not an enum, so I wrote out the string name values in the description of the object.
2. Copied the Colorant String values into the description of the object.
3. Issue 37 -*The LineStatus enum isn't used by any object/attribute. Should ChannelStatus use LineStatus, instead of EntStatus?*

15.5. ChannelType (enum type 2)

Value	Enum symbolic name	Description - becomes a MIB/MIF comment
1	Other	
2	Unknown	
3	chan Serial Port	
4	chan Parallel Port	
5	chan IEEE1284 Port	
6	chan SCSI Port	
7	chan PAP	Apple Printer Access Protocol
8	chan LPD Server	
9	chan RPrinter	Novell
10	chan Pserver	Novell
11	chan port 9100	

12	chan AppSocket	Adobe 9101
13	chan FTP	InterNet File Transfer Protocol, RFC ????, "PUT" to printer
14	chan TFTP	to printer
15	chan DLC LLC Port	
16	chan IBM 3270	IBM
17	chan IBM 5250	IBM
18	chan Fax	
19	chan IEEE1394	
20	chan Transport 1	port 35
21	chan CPAP	port 170
22	chan IPC	?
23	chan OLE	Object Linking and Embedding?, Microsoft
24	chan Named Pipe	
25	chan PCPRINT	Banyan
26	chan OS2 LAN Server	OS/2 LAN Server, IBM
27	chan DPMF	Distributed Print Management Framework, IBM

15.6. IntpLangFam - Interpreter Language Family (enum type 2)

Value	Enum symbolic name	Description - becomes a MIB/MIF comment
1	other	
2	unknown	
3	lang PCL	Printer Control Language, or Printer Command Language Starting with version 5, HP-GL/2 is part of PCL. Hewlett-Packard Co.
4	lang HPGL	Hewlett-Packard Graphics Language. Originally used to control pen movements on plotters. Includes many device specific commands. Hewlett-Packard Co.
5	lang PJL	Peripheral Job Language. Appears in the data stream between data intended for a page description language. Hewlett-Packard Co.
6	lang PostScript	Adobe Systems Incorporated Mountain View, CA
7	lang IPDS	Intelligent Printer Data Stream Bi-directional print data stream for documents consisting of data objects (text, image, graphics, bar codes), resources (fonts, overlays) and page, form and finishing instructions. Facilitates system level device control, document tracking and error recovery throughout the print process. Pennant Systems, IBM
8	lang PPDS	IBM Personal Printer Data Stream. Originally called IBM ASCII, the name was changed to PPDS when the Laser Printer was introduced in 1989. Lexmark International, Inc.
9	lang EscapeP	
10	lang Epson	
11	lang DDIF	Digital Document Interchange Format Digital Equipment Corp., Maynard MA
12	lang Interpress	
13	lang ISO 6429	ISO 6429. Control functions for Coded Character Sets (has ASCII control characters, plus additional controls for character imaging devices.) ISO Standard, Geneva, Switzerland
14	lang line data	line-data: Lines of data as separate ASCII or EBCDIC records and containing no control functions (no CR, LF, HT, FF, etc.). For use with traditional line printers. May use CR and/or LF to delimit lines, instead of records. See ISO 10175 Document Printing Application (DPA) ISO standard, Geneva, Switzerland
15	lang MODCA	Mixed Object Document Content Architecture Definitions that allow the composition, interchange -- and presentation of final form documents as a collection of data objects (text, image, graphics, bar codes), resources (fonts, overlays) and page, form and finishing instructions. Pennant Systems, IBM
16	lang ReGIS	Remote Graphics Instruction Set, Digital Equipment Corp., Maynard MA
17	lang SCS	SNA Character String Bi-directional print data stream for SNA LU-1 mode of communications IBM

18	lang SPDL	ISO 10180 Standard Page Description Language ISO Standard, Geneva, Switzerland
19	lang Tek4014	
20	lang PDS	
21	lang IGP	
22	lang CodeV	Magnum Code-V, Image and printer control language used to control impact/dot-matrix printers. QMS, Inc., Mobile AL
23	lang DSC DSE	DSC-DSE: Data Stream Compatible and Emulation Bi-directional print data stream for non-SNA (DSC) and SNA LU-3 3270 controller (DSE) communications IBM
24	lang WPS	Windows Printing System, Resource based command/data stream used by Microsoft At Work Peripherals. Developed by the Microsoft Corporation.
25	lang LN03	Early DEC-PPL3, Digital Equipment Corp.
26	lang CCITT	
27	lang QUIC	QUIC (Quality Information Code), Page Description Language for laser printers. Included graphics, printer control capability and emulation of other well-known printer languages. QMS, Inc. Mobile, AL
28	lang CPAP	Common Printer Access Protocol Digital Equipment Corp. Maynard MA
29	lang DEC PPL	Digital ANSI-Compliant Printing Protocol (DEC-PPL) Digital Equipment Corp. Maynard MA
30	lang simple text	simple-text: character coded data, including NUL, CR , LF, HT, and FF control characters. See ISO 10175 Document Printing Application (DPA) ISO standard, Geneva, Switzerland
31	lang NPAP	Network Printer Alliance Protocol IEEE 1284.1
32	lang DOC	Document Option Commands, Appears in the data stream between data intended for a page description language. QMS, Inc. Mobile, AL
33	lang imPRESS	imPRESS, Page description language originally developed for the ImageServer line of systems. A binary language providing representations for text, simple graphics (rules, lines, conic sections), and some large forms (simple bit-map and CCITT group 3/4 encoded).The language was intended to be sent over an 8-bit channel and supported early document preparation languages (e.g. TeX and TROFF). QMS, Inc. Mobile, AL
34	lang Pinwriter	24 wire dot matrix printer language for USA, Europe, and Asia except Japan. More widely used in Germany, and some Asian countries than in US. NEC
35	lang NPDL	Page printer language for Japanese market. This is one of the majors in Japanese market. NEC
36	lang 201PL	Serial printer language used in the Japanese market. This is one of the dominant in Japanese market. NEC
37	lang automatic	Automatic PDL sensing. Automatic sensing of the interpreter language family by the printer examining the document content. Which actual interpreter language families are sensed depends on the printer implementation.
38	lang PAGES	Page printer Advanced Graphic Escape Set IBM Japan
39	lang LIPS	Cannon

Notes:

*Issue 36 - Is the prefix of **lang** and **chan** the proper thing to have for MIB and MIF enum value symbols, so that the proper names and acronyms won't have their first letter made lower case?*

15.7. Alert Codes

SeverityLevelAlertCode (enum level 1) 1 Other 2 Unknown 3 Critical 4 Warning		AlertGroupCode (enum type 1) 1 Other 2 Unknown 3 Input 4 Output 5 Finisher 6 Media Path 7 General Printer 8 Marker 9 System Controller 10 Interface 11 Interpreter 12 Console 13 Channel
InputAlertCode (enum type 2) 1 Other 2 Unknown 3 Cover Open 4 Cover Closed 5 Interlock Open 6 Interlock Closed 7 Configuration Change 8 Jam 9 Media Tray Missing 10 Media Size Change 11 Media Weight Change 12 Media Type Change 13 Media Color Change 14 Media Form Parts Change 15 Media Supply Low 16 Media Supply Empty	OutputAlertCode (enum type 2) 1 Other 2 Unknown 3 Cover Open 4 Cover Closed 5 Interlock Open 6 Interlock Closed 7 Configuration Change 8 Jam 9 Media Tray Missing 10 Media Tray Almost Full 11 Media Tray Full	MarkerAlertCode (enum type 2) 1 Other 2 Unknown 3 Cover Open 4 Cover Closed 5 Interlock Open 6 Interlock Closed 7 Configuration Change 8 Jam 9 Toner Empty 10 Ink Empty 11 Print Ribbon Empty 12 Toner Almost Empty 13 Ink Almost Empty 14 Print Ribbon Almost Empty 15 Waste Toner Receptacle Almost Full 16 Waste Ink Receptacle Almost Full 17 Waste Toner Receptacle Full 18 Waste Ink Receptacle Full 19 Fuser Under Temperature 20 Fuser Over Temperature 21 OPC Life Almost Over 22 OPC Life Over 23 Developer Almost Empty 24 Developer Empty
GeneralPrinterAlertCode (enum type 2) 1 Other 2 Unknown 3 Cover Open 4 Cover Closed 5 Interlock Open 6 Interlock Closed 7 Door Open 8 Door Closed 9 Power Up 10 Power Down	MediaPathDeviceAlertCode (enum type 2) 1 Other 2 Unknown 3 Cover Open 4 Cover Closed 5 Interlock Open 6 Interlock Closed 7 Configuration Change 8 Jam 9 Media Tray Missing 10 Media Tray Almost Full 11 Media Tray Full	FinisherAlertCode (enum type 2) 1 Other 2 Unknown 3 Cover Open 4 Cover Closed 5 Interlock Open 6 Interlock Closed 7 Configuration Change 8 Jam 9 Media Tray Missing 10 Supply Empty 11 Supply Low 12 Receptacle Almost Full 13 Receptacle Full

InterpreterAlertCode (enum type 2)		InterfaceAlertCode (enum type 2)		ChannelAlertCode (enum type 2)	
1	Other	1	Other	1	Other
2	Unknown	2	Unknown	2	Unknown
3	Configuration Change	3	Configuration Change	3	Configuration Change
4	Memory Increase				
5	Memory Decrease				
6	Cartridge Added				
7	Cartridge Deleted				
8	Resource Added				
9	Resource Deleted				
10	Resource Unavailable				

16. Appendix A - Glossary of terms

Alert -- a reportable event for which there is an entry in the alert table

Attribute - a data item that has a name, a syntax, and a value (DMTF usage). Analogous to the SNMP MIB concept of object.

Bin -- an output sub-unit which may or may not be removable

Channel -- A term used to describe a single source of data which is presented to a printer. The model that we use in describing a printer allows for an arbitrary number of channels. Multiple channels can exist on the same physical port. This is commonly done over EtherNet ports where EtherTalk, TCP/IP, and SPX/IPX protocols can be supplying different data streams simultaneously to a single printer on the same physical port.

Component -- a complete product, such as a printer. (DMTF usage)

Control Language - a data syntax or language for controlling the printer through the print data channel.

Critical Alert -- an alert triggered by an event which leads to a state in which printing is no longer possible; the printer is stopped

Description -- information about the configuration and capabilities of the printer and its various sub-units

DMTF - Desktop Management Task Force - a group of companies that has defined a management framework for PCs and other hosts for local management of hardware and software with automatic interworking with SNMP for remote management

DPA - ISO 10175 Document Printing Application standard. A standard for a client server protocol for a print system, including (1) submitting print jobs to and (2) managing print jobs in a spooler

Event - a state change in the printer

Group -- a collection of attributes or objects that represent a type of sub-unit of the printer

IANA - Internet Registration Authority. See RFC ????

Idempotent -- The notion of idempotent is that SNMP packets are datagrams and do not have guaranteed delivery. We would like to choose commands which yield the same result when executed multiple times as when executed once. For instance a switch like the foot switch that changes from high beams to low beams is not idempotent. If the command is received multiple times the result may be different than if the command was received a single time. In the SNMP world preferred commands would be "set lights to high beam" and "set lights to low beam". These commands yield predictable results when executed perhaps multiple times. A command like "press foot toggle switch", when executed an unknown number of times, yields an indeterminate result.

Input -- a tray or bin from which instances of the media are obtained and fed into the Media Path

Interpreter - the embodiment of an algorithm that processes a data stream consisting of a Page Description Language (PDL) and/or a Control Language.

Localization -- the specification of human language, country, and character set needed to present information to people in their native languages.

Management Application (a.k.a. Manager) -- a program which queries and controls one or more managed nodes

Management Station -- a physical computer on which one or more management applications can run

Media Path -- the mechanisms that transport instances of the media from an input, through the marker, possibly through media buffers and duplexing pathways, out to the output with optional finishing applied. The inputs and outputs are not part of the Media Path.

MIB - Management Information Base - the specification for a set of management objects to be managed using SNMP or other management protocol; also an instance of the data for such a set

MIF - Management Information Format - the specification for a set of management attributes to be managed locally on a PC or other host using the Desktop Management Task Force (DMTF) Management Framework.; also an instance of the data for such a set

Non-critical Alert -- an alert triggered by a reportable event which does not lead to a state in which printing is no longer possible; such an alert may lead to a state from which printing may no longer be possible in the future, such as the low toner state or the alert may be pure informational, such as a configuration change at the printer.

Object - a data item that has a name, a syntax, and a value (IETF SNMP usage). Analogous to the DMTF MIF concept of attribute

Object/attribute - a term used to mean an object in a MIB and an attribute in a MIF

Output -- a bin or stacker which accepts instances of media that have been processed by a printer

Page Description Language (PDL) - a data syntax or language for the electronic representation of a document as a sequence of page images.

Printer -- a physical device that takes media from an input source, produces marks on that media according to some page description or page control language and puts the result in some output destination, possibly with finishing applied.

Printing -- the entire process of producing a printed document from generation of the file to be printed, choosing printing properties, selection of a printer, routing, queuing, resource management, scheduling, and finally printing including notifying the user

Reportable event -- an event that is deemed of interest to a management station watching the printer

Status -- information regarding the current operating state of the printer and its various sub-units. This is an abstraction of the exact physical condition of the printer.

Sub-mechanism -- a distinguishable part of a sub-unit

Sub-unit -- a part of the printer which may be a physical part, such as one of the input sources or a logical part such as an interpreter.

Tray -- an input sub-unit which is typically removable

Visible state -- that portion of the state of the printer that can be examined by a management application

17. Appendix B - Media Size Names

This appendix lists the standardized media size names from ISO/IEC 10175 Document Printing Application (DPA). Management application developers are encouraged to use these media sizes in order to display a media size name to the user, since the names are more likely to be familiar to them than the numerical sizes returned as the values of **prtInputMediaDimXFeedDir** and **prtInputMediaDimFeedDir**.

String name	Description
other	
unknown	
na-letter or letter	North American letter size: 8.5 by 11 inches
na-legal or legal	North American legal size: 8.5 by 14 inches
na-10x13-envelope	North American 10x13 size: 10 by 13 inches
na-9x12-envelope	North American 9x12 size: 9 by 12 inches
na-number-10-envelope	North American number 10 business envelope
na-7x9-envelope	North American 7x9 size: 7 by 9 inches
na-9x11-envelope	North American 9x11 size: 9 by 11 inches
na-10x14-envelope	North American 10x14 size: 10 by 14 inches
na-number-9-envelope	North American number 9 business envelope
na-6x9-envelope	North American 6x9 size: 6 by 9 inches
na-10x15-envelope	North American 10x15 size: 10 by 15 inches
a	engineering A size
b	engineering B size
c	engineering C size
d	engineering D size
e	engineering E size
iso-a0	ISO A0 size: 841 mm by 1189 mm
iso-a1	ISO A1 size: 594 mm by 841 mm
iso-a2	ISO A2 size: 420 mm by 594 mm
iso-a3	ISO A3 size: 297 mm by 420 mm
iso-a4	ISO A4 size: 210 mm by 297 mm
iso-a5	ISO A5 size: 148 mm by 210 mm
iso-a6	ISO A6 size: 105 mm by 148 mm
iso-a7	ISO A7 size: 74 mm by 105 mm
iso-a8	ISO A8 size: 52 mm by 74 mm
iso-a9	ISO A9 size: 37 mm by 52 mm
iso-a10	ISO A10 size: 26 mm by 37 mm
iso-b0	ISO B0 size: 1000 mm by 1414 mm
iso-b1	ISO B1 size: 707 mm by 1000 mm
iso-b2	ISO B2 size: 500 mm by 707 mm
iso-b3	ISO B3 size: 353 mm by 500 mm
iso-b4	ISO B4 size: 250 mm by 353 mm
iso-b5	ISO B5 size: 176 mm by 250 mm
iso-b6	ISO B6 size: 125 mm by 176 mm
iso-b7	ISO B7 size: 88 mm by 125 mm
iso-b8	ISO B8 size: 62 mm by 88 mm
iso-b9	ISO B9 size: 44 mm by 62 mm
iso-b10	ISO B10 size: 31 mm by 44 mm
iso-c3	ISO C3 size: 229 mm by 324 mm
iso-c4	ISO C4 size: 229 mm by 324 mm
iso-c5	ISO C5 size: 162 mm by 229 mm
iso-c6	ISO C6 size: 162 mm by 229 mm
iso-designated	ISO Designated Long size: 110 mm by 220 mm
jis-b0	JIS B0 size
jis-b1	JIS B1 size
jis-b2	JIS B2 size
jis-b3	JIS B3 size
jis-b4	JIS B4 size
jis-b5	JIS B5 size
jis-b6	JIS B6 size
jis-b7	JIS B7 size

jis-b8	JIS B8 size
jis-b9	JIS B9 size
jis-b10	JIS B10 size

18. Appendix C -Media Names

This appendix lists the standardized media names from ISO/IEC 10175 Document Printing Application (DPA). Printer vendors and system mangers are encouraged to use these media names as values of the **prtInputMediaName** object/attribute in order to display a media name to management applications. Using standard media names will mean that a single management application dealing with printers from different vendors and under different system mangers will tend to use the same names for the same media. Furthermore, different management applications dealing with the printers from the same or different printer vendors will show their users the same names, thereby reducing user confusion. Of course printer vendors and system managers will need to invent additional names to cover non-standard media.

Simple Name	Descriptor Text
iso-a4-white	Specifies the ISO A4 white medium with size: 210 mm by 297 mm as defined in ISO 216
iso-a4-coloured	Specifies the ISO A4 coloured medium with size: 210 mm by 297 mm as defined in ISO 216
iso-a4-transparent	Specifies the ISO A4 transparent medium with size: 210 mm by 297 mm as defined in ISO 216
iso-a3-white	Specifies the ISO A3 white medium with size: 297 mm by 420 mm as defined in ISO 216
iso-a3-coloured	Specifies the ISO A3 coloured medium with size: 297 mm by 420 mm as defined in ISO 216
iso-a5-white	Specifies the ISO A5 white medium with size: 148 mm by 210 mm as defined in ISO 216
iso-a5-coloured	Specifies the ISO A5 coloured medium with size: 148 mm by 210 mm as defined in ISO 216
iso-b4-white	Specifies the ISO B4 white medium with size: 250 mm by 353 mm as defined in ISO 216
iso-b4-coloured	Specifies the ISO B4 coloured medium with size: 250 mm by 353 mm as defined in ISO 216
iso-b5-white	Specifies the ISO B5 white medium with size: 176 mm by 250 mm as defined in ISO 216
iso-b5-coloured	Specifies the ISO B5 coloured medium with size: 176 mm by 250 mm as defined in ISO 216
jis-b4-white	Specifies the JIS B4 white medium with size: ??? mm by ??? mm as defined in JIS ????
jis-b4-coloured	Specifies the JIS B4 coloured medium with size: ??? mm by ??? mm as defined in JIS ????
jis-b5-white	Specifies the JIS B5 white medium with size: ??? mm by ??? mm as defined in JIS ????
jis-b5-coloured	Specifies the JIS B5 coloured medium with size: ??? mm by ??? mm as defined in JIS ????

The following standard values are defined for North American media:

na-letter white	Specifies the North American letter white medium with size: 8.5 inches by 11 inches
na-letter coloured	Specifies the North American letter coloured medium with size: 8.5 inches by 11 inches
na-letter transparent	Specifies the North American letter transparent medium with size: 8.5 inches by 11 inches
na-legal white	Specifies the North American legal white medium with size: 8.5 inches by 14 inches
na-legal coloured	Specifies the North American legal coloured medium with size: 8.5 inches by 14 inches

The following standard values are defined for envelopes:

iso-b5-envelope	Specifies the ISO B5 envelope medium with size: 176 mm by 250 mm as defined in ISO 216 and ISO 269
iso-b4-envelope	Specifies the ISO B4 envelope medium with size: 250 mm by 353 mm as defined in ISO 216
iso-c4-envelope	Specifies the ISO C4 envelope medium with size: 229 mm by 324 mm as defined in ISO 216 and ISO 269
iso-c5-envelope	Specifies the ISO C5 envelope medium with size: 162 mm by 229 mm as defined in ISO 269
iso-designated-long-envelope	Specifies the ISO Designated Long envelope medium with size: 110 mm by 220 mm as defined in ISO 269
na-10x13-envelope	Specifies the North American 10x13 envelope medium with size: 10 inches by 13 inches
na-9x12-envelope	Specifies the North American 9x12 envelope medium with size: 9 inches by 12 inches
na-number-10-envelope	Specifies the North American number 10 business envelope medium with size: 4.125 inches by 9.5 inches
na-7x9-envelope	Specifies the North American 7x9 inch envelope
na-9x11-envelope	Specifies the North American 9x11 inch envelope
na-10x14-envelope	Specifies the North American 10x14 inch envelope
na-number-9-envelope	Specifies the North American number 9 business envelope
na-6x9-envelope	Specifies the North American 6x9 inch envelope
na-10x15-envelope	Specifies the North American 10x15 inch envelope

The following standard values are defined for the less commonly used media (white-only):

iso-a0-white	Specifies the ISO A0 white medium with size: 841 mm by 1189 mm as defined in ISO 216
iso-a1-white	Specifies the ISO A1 white medium with size: 594 mm by 841 mm as defined in ISO 216
iso-a2-white	Specifies the ISO A2 white medium with size: 420 mm by 594 mm as defined in ISO 216
iso-a6-white	Specifies the ISO A6 white medium with size: 105 mm by 148 mm as defined in ISO 216
iso-a7-white	Specifies the ISO A7 white medium with size: 74 mm by 105 mm as defined in ISO 216
iso-a8-white	Specifies the ISO A8 white medium with size: 52 mm by 74 mm as defined in ISO 216

iso-a9-white	Specifies the ISO A9 white medium with size: 39 mm by 52 mm as defined in ISO 216
iso-10-white	Specifies the ISO A10 white medium with size: 26 mm by 37 mm as defined in ISO 216
iso-b0-white	Specifies the ISO B0 white medium with size: 1000 mm by 1414 mm as defined in ISO 216
iso-b1-white	Specifies the ISO B1 white medium with size: 707 mm by 1000 mm as defined in ISO 216
iso-b2-white	Specifies the ISO B2 white medium with size: 500 mm by 707 mm as defined in ISO 216
iso-b3-white	Specifies the ISO B3 white medium with size: 353 mm by 500 mm as defined in ISO 216
iso-b6-white	Specifies the ISO B6 white medium with size: 125 mm by 176 mm as defined in ISO 216
iso-b7-white	Specifies the ISO B7 white medium with size: 88 mm by 125 mm as defined in ISO 216
iso-b8-white	Specifies the ISO B8 white medium with size: 62 mm by 88 mm as defined in ISO 216
iso-b9-white	Specifies the ISO B9 white medium with size: 44 mm by 62 mm as defined in ISO 216
iso-b10-white	Specifies the ISO B10 white medium with size: 31 mm by 44 mm as defined in ISO 216
jis-b0-white	Specifies the JIS B0 white medium
jis-b1-white	Specifies the JIS B1 white medium
jis-b2-white	Specifies the JIS B2 white medium
jis-b3-white	Specifies the JIS B3 white medium
jis-b6-white	Specifies the JIS B6 white medium
jis-b7-white	Specifies the JIS B7 white medium
jis-b8-white	Specifies the JIS B8 white medium
jis-b9-white	Specifies the JIS B9 white medium
jis-b10-white	Specifies the JIS B10 white medium

The following standard values are defined for engineering media:

a	Specifies the engineering A size medium
b	Specifies the engineering B size medium
c	Specifies the engineering C size medium
d	Specifies the engineering D size medium
e	Specifies the engineering E size medium

19. Appendix D - Roles of Users

19.1. Background

The need for Role Models stemmed in large part from the need to understand the importance any given managed object under consideration for inclusion in the specification. Many times the presence or nature of a particular proposed object would be debated within the group; the debate would typically end when one or more persons would describe the potential usage for the object, usually in terms of a "live" person operating in some target environment.

Steve Zilles (Adobe) first mentioned that he had considered this general problem and had come up with a short list of categories by which the group can evaluate the relative utility of a proposed object. The list Steve described was:

- User
- Trained Operator
- Service

Upon further examination of the overall problem I found it useful to expand the list of categories, as well as attempt to define a basic set of "requirements areas" that can help define the basic nature of each category.

Every concept needs a name, and this concept is no different. For lack of better alternatives, I refer to these categories as "Role Models" in this document. This name was chosen in light of the fact that many times we try to find a "person" (or similar entity) for which the use of a proposed object is targeted. (I resisted the temptation to use the term "Usage Models," as I felt the term was too generic in nature.)

In presenting the initial list of Role Models, the initial set of "requirements areas" are presented, followed by the set of Role Model definitions. Finally, a simple matrix is presented in which Role Models and requirements areas are cross-compared.

It should be emphasized at this point that all of this is proposed as initial information for further discussion. No doubt major changes will be proposed by members of the group as time goes on.

19.2. Proposed Print System Requirements Areas

In examining the various kinds of information and operations surrounding printers and printing systems, the following list of "requirement areas" is proposed as a "check list" of needs for the various Role Models:

Printer job state - Determine the status of a job without a printer.

Printer capabilities - Determine the current capabilities of a printer, for example, the available media sizes, two-sided printing, a particular type of interpreter, etc.

Printer job submission - Submit a print job to a printer.

Printer job removal - Remove a job from a printer.

Notification of events - Receive notification of the existence of a defined printer event. An event can be of many types, including warnings, errors, job stage completion (e.g., "job done"), etc.

Printer configuration - Query the current configuration of a printer.

Printer consumables - Determine the current state of any and all consumables within a printer.

Print job identification - Determine the identification of a job within a printer.

Internal printer status - Determine the current status of the printer.

Printer identification - Determine the identify of a printer.

Printer location - Determine the physical location of a printer.

Local system configuration - Determine various aspects of the current configuration of the local system involved with the operation of a printer.

These "requirements" cover a large spectrum of requirements surrounding the operation of a printer in a network environment. This list is by no means complete, but serves as a starting point for assessing major requirements of the various Role Models described below.

19.3. Proposed Role Models

Following is a proposed list of "Role Models" to be used in evaluating the requirements for any given object defined within the Printer MIF. Note that the keyword enclosed in parentheses represents an abbreviation for the particular Role Model in the matrix described later in this document.

User (USER) - A person or application that submits print jobs to the printer; typically viewed as the "end user" within the overall printing environment.

Operator (OP) - A person responsible for maintaining a printer on a day-to-day basis, including such tasks as filling empty media trays, emptying full output trays, replacing toner cartridges, etc.

Technician (TECH) - A person responsible for repairing a malfunctioning printer, performing routine preventive maintenance, and other tasks that typically require advanced training on the printer internals. An example of a "technician" would be a manufacturer's Field Service representative, or other person formally trained by the manufacturer or similar representative.

System Manager (MGR) - A person responsible for configuration and troubleshooting of components involved in the overall printing environment, including printers, print queues and network connectivity issues. This person is typically responsible for ensuring the overall operational integrity of the print system components, and is typically viewed as the central point of coordination among all other Role Models.

Help Desk (HELP) - A person responsible for supporting Users in their printing needs, including training Users and troubleshooting Users' printing problems.

Asset Manager (AM) - A person responsible for managing an organizations printing system assets (primarily printers). Such a person needs to be able to identify and track the location of printing assets on an ongoing basis.

Capacity Planner (CP) - A person responsible for tracking the usage of printing resources on an ongoing basis. An optional related activity might be to acquire printing resource utilization information for the purposes of charging Users for resources used.

Installer (INST) - A person or application responsible for installing or configuring printing system components on a local system.

The purpose of these Role Models is to evaluate the relative merit of any given managed object. Whenever a managed object is proposed for inclusion into the specification, discussion on its expected value should be geared around which Role Models benefit from its presence and operation.

19.4. Matrix of Requirement Areas and Role Models

To better understand the relationship between the set of defined "Requirements Areas" and the various "Role Models," the following matrix is offered.

It is important to recognize that many of the requirements areas will appear to be applicable to many of the Role Models. However, when considering the actual context of a requirement area, it is very important to realize that often the actual context of a requirement is such the Role Model can change.

For example, it is obvious that a "System Manager" must be able to submit print jobs to a printer; however, when submitting a print job a person identified as a "System Manager" is actually operating in the context of a "User" in this case; hence, the requirement to submit a print job is not listed as a requirement for a System Manager.

Conversely, while a "User" must be able to remove a job previously submitted to a printer, an "Operator" is often expected to be able to remove any print job from any printer; hence, print job removal is a (subtly different) requirement for both "User" and an "Operator" Role Models.

That being said, I'm sure you'll find some inconsistencies in the following matrix, depending on your particular interpretations of the various requirements areas.

Role Models								
Requirement Area	USER	OP	TECH	MGR	HELP	AM	CP	INST
Print job status	xx	xx	xx	xx	xx			
Printer capabilities	xx			xx	xx			
Print job submission	xx							
Print job removal	xx	xx						
Notification of events		xx	xx					
Printer configuration				xx				xx
Printer consumables		xx					xx	
Print job identification		xx		xx	xx		xx	
Internal printer status		xx	xx	xx				
Printer identification		xx		xx	xx	xx	xx	xx
Printer location							xx	
Local system configuration				xx				xx

20. Appendix E - Participants

The following people attended at least one meeting of the Printer Working Group meeting; many attended most meetings.

Azmy Abouased - Compaq
Avi Basu - HP
Kerry Bott - Intel
Ted Brunner - Tektronix
Jeff Case - SNMP Inc.
Rong Chang - IBM
Andy Davidson - Tektronix
Jack Demcak - Vadtech
Andria Demetroulakos - Digital Products
Mike Evans - ESI
Richard Everman - uci.edu
Neal Fischer - Fujitsu
Joseph Flick - HP
Rod Gerhart - Ricoh
Christine Gressley - University of Illinois
Joel Gyllenskog - HP
Tom Hastings - DEC
Tim Hathaway - Pacific Data Products
Bob Herriot - SUN
Jeff Johnson - Cisco
Jeff Johnson - Microsoft
Barry Kelman - Microsoft
Charles Kimber - Dataproducts
Theodore Kearley - QMS
Andrew Knutsen - SCO
Peter Leunig - Leunig GmbH
Harry Lewis - IBM Pennant Systems
Bill Lott - QMS
Mike MacKay - Xerox
Jay Martin - Underscore
Mike Mayes - Brother
Kevin McBride - Underscore
Stan McConnell - XEROX
Michael Moore - Ricoh
Rudy Nedved - CMU Computer Science Dept.
Pete Neergaard - CMU Printing Group
Bill Norton - merit.edu
Ron Norton - Printronix
Roman Orzol - Okidata
Alan Perelman - Emulex
Noga Prat - Intel
Dave Roach - Unisys
Marshall Rose
Mike Scanlon - FTP Software
Avi Schlank - Canon
Ron Smith - TI
Larry Stein - Farpoint
Koji Tashiro - NEC Technologies
Jody Terrill - Extended Systems
Chris Thomas - Intel
Mike Timperman - Lexmark
Randy Turner - QMS
Bill Wagner - Digital Products
Tim Wells - Microsoft
Craig Whittle - Compaq
Don Wright - Lexmark
Steve Zilles - Adobe
Jim Zuber - Genoa

21. Appendix Y - Issues

This appendix lists the issues that have been raised during the meetings and in between meetings. The issues listed under each MS-WORD table under NOTES: are additional issues which are not included here.

1) Is the media group optional?

   Leave MediaExtended as optional, media information in the InputTable will remain mandatory.

2) Will the MIB/MIF support multiple consoles?

   No, do with private extensions, a single "console" that contains the data for all the consoles, or some other means.

3) Is General Printer a table?

   No, support Virtual printers through some other means such as private extensions, etc.

4) How does "character set" tie into "channel"?

   There was no identified need to tie character set to channels. Character Set is more closely tied to the interpreter or console and those objects are tied to localization and character set settings.

5) Static versus dynamic - how efficient will this be?

   Because there is no distinction between static and dynamic information in SNMP, the printer must implement all the information so the agent can report it.

6) The MIB/MIF consist of 155 objects, 98 being mandatory - is that too many?

   OPEN --- Defer for final review.

7) Should we endorse some type of interlock method for all the writable objects?

   OPEN --- One of the reasons it remains open is that there is no defined method for this function for DMTF.

8) Should we use separate community names or additional index to support multiple printers being managed through a printer server?

   Use community names with predictable "numbering" for each device. Randy Turner (QMS) will write up a proposal for inclusion in the RFC.

9) Should paper size be writable?

   Yes, to tell a printer that can't detect paper size what size paper is loaded.

10) Should units be writable?

    Yes, but manufacturers may choose to only implement R/O access. Persistence across power cycles may vary from manufacturer to manufacturer.

11) Should size units be only in millimeters?

    No, but we will reduce the number of units wherever possible.

12) How should out of range sets be handled - reject or snap to the nearest?

    Snap to the nearest integer and reject an invalid ENUM. The data returned in the SNMP packet will indicate the value of the integer snapped to.

13) Are "Input - dimensions extended" mandatory?

    This will remain at best optional; possibly could be removed.

14) In what units is printable area reported?

    .0001 inches or .001 mm

15) Should media type be a string or an enum?

    String, recommend using ISO DPA, ISO SPDL names where possible.

16) How do we associate media with inputs, outputs with finishers, etc.?

    Input: The only media that is described is the media in an input tray and it is described there.

    Outputs: OPEN - Harry Lewis to make a proposal

17) Do margins (or imagable area) need to be PDL specific?

    No, but this object may not be needed.

18) Does "other" have a meaning for objects which normally have an integer value?

    Yes - potential meanings are defined with the object where it applies.

19) Is "unknown" the same as "uninitialized" and if not how do you tell the difference?

    Possibly, if the object's value is "unknown" failing when attempting to write to it means the object is truly unknown. If writing is successful, the object's value was just initialized.

20) Should default interpreter be by channel or overall?

    OVERALL. There is a hierarchy used to determine the interpreter:

    1) If contained in the data stream, used that interpreter

    2) Otherwise use the default for the channel

    3) Otherwise use the default for the interpreters

    4) Otherwise the default interpreter is undefined or manufacturer specific

21) How do we handle the "parallel in, serial out" channel?

    Ignore it

22) Is "console width" "string width" or "screen width"?

    Closed by addressing issue #23.

23) Is "number of lines" and "number of characters" of the console mean physically or logical?

    They refer to the physical size of the console.

24) Which strings are localized and controlled by "current localization?"

    Spec has been updated to reflect which strings are localized by **prtGeneralCurrentLocalization**, which strings are localized by **prtConsoleLocalization** and which strings are not localized. String not specifically stated to be localized should not be localized as that data may be used to make decisions.

25) Do we need an "ID" object for each table "index?"

    OPEN

26) Are all consoles using the same language?

    We only support one console.

27) Should we enable or disable a channel as a whole or provide a way to selectively enable and disable "data" and "control" flows on a channel?

    No convincing argument for separation of control and data. Therefore, the answer is no.

28) Is the consistent use of 1,2 (for enums) and -1, -2 (for integers) to represent "other" and "unknown" consistent with SNMP "tradition?"

    There is a tradition of having certain distinguished values but those values often vary from RFC to RFC. Consistent of use of these values within the PRINTER MIB should be OK.

29) The prefix for SNMP MIB object names is **prt**; do the DMTF MIF attribute names have to have this same prefix or can they avoid having such a prefix, since MIF names are quoted strings which can be any case and because the MIF names are displayed directly to users, correct?

    OPEN Need DMTF experts help here.

30) Should the DMTF MIF attribute names include "-" when the word in naturally hyphenated, such as in the Interpreter Language **line-data**, or include "/" such as in the channel **OS/2 LAN Manager**?

31) Need upper bounds on each object of type String.

OPEN I've put in some straw man values. Mostly 255, since SNMP RFC indicates a problem if DisplayString is greater than 255 octets. Also ISO DPA uses 255 for its descriptor attributes. 63 for standardized string values, so that can get 31 characters in a two-octet character set. 31 for Version and LangLevel, so can get 15 characters in a two-octet character set.

32) The DMTF Component ID attributes seem to be quite different from the System Group in the MIB-II RFC 1213. Is this difference a problem? How will the MIF to MIB translator handle this difference?

OPEN We need DMTF experts help here.

33) Should we change the name from "Skins" to "Doors" to be more PC?

OPEN

34 Should MediaType and MediaColor string value names be localized?

OPEN - but probably yes.

35 Is a constraint of 63 octets enough for translating MediaType and MediaColor string values into all languages and character sets (two octet character sets use up two octets per character) sufficient?

OPEN

36 Is the prefix of **lang** and **chan** the proper thing to have for MIB and MIF enum value symbols, so that the proper names and acronyms won't have their first letter made lower case?

OPEN We need DMTF experts help here.

37 The LineStatus enum isn't used by any object/attribute. Should ChannelStatus use LineStatus, instead of EntStatus?

OPEN

38 What objects from the System Group (MIB-II RFC 1213), Host Resources MIB (RFC 1514), and Interfaces Group of MIB-II (RFC 1573) shall be listed in the Printer MIB as being required for conformance to the Printer MIB?

OPEN Need some SNMP experts help here.

39 Are all enums of the type that we want?

a. Why is MrkSupplyType enum type 3, instead of type 2? Don't we want to review them?

b. Why is ResetType enum type 3, instead of type 2? Don't we want to review them?

c. Why is ColorantRole enum type 1, instead of type 2? Do we really want to do another RFC when a new ColorantRole, beside process and spot gets invented?

d. Why is ChannelState enum type 1, instead of type 2? Suppose we do want to control the Control Language separately from the Page Description Language in the future (or an implementor wants to)? Do we really want to do another RFC?

e. Why is AlertGroupCode enum type 1, instead of type 2? For example, suppose the Finisher Working Group decides to have two groups, not one for Finishing?

40. Are the names of the objects/attributes what we want, especially since the MIF names are diaplayed to users directly (assume that for the MIF, each word in an attribute name is space separated). Some of the names are quite long, containing a lot of words which have been abbrievated beyond many DMTF user's recognition. Should we remove some of the words and make the remaing words be full words, except for obvious abbrievations, like Max and Lang?

OPEN - Use the index to make such global decisions about names.

22. Appendix Z - Change History

This appendix list the changes made to each version of the specification, listed in reverse chronological order.

22.1. Changes from Version 0.7, 27-Apr to Version 0.1, 15-May

1. Changed the title from Management Object Specification to Common MIB/MIF Specification to reflect the joint nature of this specification.
2. Changed the file name from **pmi_obj*N*.doc** to **prtspec*N*.doc** and **prtspec*N*.ps** now that the model text has been added so that this document is a complete specification. Also we were running out of single digit fraction version numbers. The corresponding files with revisions marks (r) left in have the file name form: **prtspec*N*.dcr** and **prtspec*N*.psr**. The files for producing simple-text (t) have the file name form: **prtspec*N*.dct** and **prtspec*N*.txt**
3. Added Printer Model Version 0.4, dated 7-May.
4. Merged all of the alert specification for objects from the model document into the object descriptions.
5. Replaced the WORD graphic package figures with simple ASCII pictures so that the pictures may be extracted directly into the MIB and MIF specs.
6. Ordered the Model discussion and the group descriptions to be the same.
7. Updated the Interpreter Group with Joel's Version 8 (intepreters.08), dated 10-May.
8. Added all-lower case "lang " and "chan " words in front of the InterpreterLangFamily and Channel enums, so that the first letter lower case in SNMP would not lower case the proper names that most of these enum symbols are.
9. Changed all object names that contained **CrossFeed** or **AcrossFeed** to **XFeed** to agree with the same change to the Interpreter group and to name cross feed the same for all objects.
10. Changed the template to be **prtspec.dot** and added the following styles to help make it easier to generate the spec and to produce the simple-text form suitable for producing the ASCII MIB and MIF specs:

Table Heading	- first line of a table
Table Legend	- second line of tables that list column headings: Object Name, Access, Type, Value ...
Table	- lines in tables after the first
Syntax	- lines of fixed pitch examples
ascii pic	- ASCII pictures, so that they can be extracted into MIB and MIF specs
enumlev1	- first level list items
enumlev2	- second level list items
heading 1	- first level headers - for generating table of contents
heading 2	- through 4 for 2nd to 4th level headers

11. Removed all special characters (-, /, :, and ,) from enum symbols, so that they will follow SNMP V2 rules.
12. Uppercased the first letter of each word in enum symbols. Kept all caps acronyms all caps.
13. Left a space between each word in enum symbols, since spaces are allowed in DMTF enum symbols; the equivalent IETF SNMP enumerated value symbol will have the first letter of the first word lower cased and the spaces squeezed out.
14. Changed enum symbol "1 to n" to "One To N" , since enum symbols can't start with a digit.
15. Added Steve's write-up on read-write objects/attributes that are writeable only for printer implementation that cannot sense the value itself.
16. Added **prtGeneral** to all references to **General** object from other groups (e.g., **prtGeneralCurrentLocalization**) to follow the conventions of using the full object/attribute name for inter group references.

17. Added a constraint maximum length to all String data types. Most are 255, as indicated in SNMP RFC as causing problems if greater than 255. Also ISO DPA uses 255 octets as its constraint for descriptor attributes. Standardized string value object/attributes, such as MediaType and MediaColor, I put as 63, so that can represent 31 characters in a two-octet character set in any language. 31 for Version and LangLevel, so can get 15 characters in a two-octet character set.
18. Added the following terms to the Glossary: Attribute, Channel, Control Language, DMTF, DPA, Event, Idempotent, Interpreter, MIB, MIF, Object, Object/Attribute, and Page Description Language.
19. Added "**s**" to the names of objects/attributes of type Counter32, as per SNMP V2 RFC, except the ones that ended in **Count**.
20. Added participants as an Appendix.
21. Added Jay Martin's user roles paper as an Appendix.
22. Added the enum type numbers to the tables at the end, so that its easier to compare the types.
23. Checked all references to enum symbols and that the enum types agree for types that are multiply referenced. Made YesNoOnOff agree to be enum (1).
24. Changed the file name to **prtspec*N*.dcr** for the version with revision marks (r). The **prtspec*N*.doc** version is the same, but with the revions accepts, so that there are no revision marks.
25. Added the string values for **MediaType**, **MediaColor**, and **MarkerColorantValue** from ISO DPA and ISO SPDL right into the descriptions of these objects/attributes.
26. Deleted the redundant **Marker** and **Colorant** from the MarkerSupplies and MarkerColorant table objects.
27. Added the index and indexed all objects/attributes, except Finishing, because they may be done by another working group.
28. Added a last sentence to each object/attribute description that is an enum: This value is an enum of type x. Then the text descriptoin can be used directly in the MIB and MIF specs and MIB and MIF sources, scine the enum(n) notation in our MS-WORD tables isn't present in MIB and MIF syntax.
29. Added an Appendix of standard media names from ISO DPA.
30. Replaced all WORD DRAW figures with ASCII text pictures, since the text pictures are required for MIB and MIF specs.
31. Changed the headers and footers to agree with IETF requirements for Internet drafts and RFCs.

22.2. Changes from Version 0.6, 26-Apr, to Version 0.7, 27-Apr

1. Added console lock
2. Changed printable area to margins and moved to marker
3. Worked through all notes and removed after resolving the issues.
4. Updated issues list based upon second day of meeting (April 27, 1994)
5. Added finisher consumables units

22.3. Changes from Version 0.2, 25-Apr, to Version 0.6, 26-Apr

1. Cleaned-up formatting
2. Updated issues list based upon first day of meeting in April.
3. Changed version number to align with original PMI_OBJ series of documents.
4. Hide Tom Hastings editorial comments on page 2.
5. This version was only distributed at the April IETF/DMTF meeting.

22.4. Changes from Version 0.1, 23-Apr, to Version 0.2, 25-Apr

1. Added Appendix Y, List of Issues from the ftp-boi.external.hp.com directory.

2. Replaced Interpreter Language Family enum table with the contents of the Interpreter Spec, so that the comments are included as well for each enum value. Now this specification will be the only copy of the enum values assigned.

22.5. Changes to create Version 0.1, 23-Apr

1. Started with Version 0.5 of the Object Summary that Don edited with agreements reached at the March Seattle meeting.

2. Copied in the descriptions from the following specifications that the working group had reviewed: and had been posted in the ftp-boi.external.hp.com server:

number	file name	date	groups
1	general/pmi_obj5.doc	22-Apr	all
2	channel_class/channel3.mif	19-Apr	Channel
3	channel2 mail	28-Mar	Console
4	source_dest_class/pmi_inp_outp_V0.3.txt	23-Feb	Input, Output, Finisher
5	interpreters_class/interpreters.05	21-Apr	Interpreter
6	localization/localization.objects	16-Mar	General
7	printer_class/prt_mif4.doc	21-Feb	Marker, MediaPath
8	general/pmi_modl.doc	30-Mar	Alerts
9	general/glossary.txt	29-Mar	Appendix

My MS-WORD for Windows 2.0a could not read the pmi_mdl3.doc, 19-Apr.

3. Changed the names of objects/attributes and enum symbols that had the word **Units** to **Unit** to be consistent with the rest that had just **Unit** Also changed **Dims** to **Dim** to be consistent with the rest that had just **Dim**.

4. Changed the terminology of the descriptions that used "units" to "unit of measure", so as to avoid confusion with hardware units and sub-units.

5. Spell checked the entire document and created a custom dictionary, **pmispec.dic**.

6. Changed the names of a few of the Language Families that had come from the ISO DPA, to agree with the names in the ISO DPA. In half of the cases, the ISO DPA committee agreed to change their names. The following changes were made in this specification: 1) LinePrinter to line-data, 2) SimpleText to simple-text.

7. Changed the Language Family enum values that had been added to the Object Summary V0.5, after the Seattle meeting to agree with the 21-Apr version of the Interpreter specification: interpreters.05.

8. Added automatic as enum 37, since the committee had agree in Seattle and SnowBird to add automatic as a PDL for which the printer determines which interpreter to use by looking at the print data itself.

9. Editorially changed the beginning of each object/attribute description to be a noun that is the value and start each with the definite article: "The" or "A".

10. Removed the terms object in the object/attribute descriptions since IETF MIBs use the term objects, while the DMTF MIF use the term attribute. Still need to change the title of the specification though, since it contains “Object”.

23. Index

AddressabilityFeedDir
in prtMarker, 30
AddressabilityUnit
in prtMarker, 30
AddressabilityXFeedDir
in prtMarker, 30
AlertCode
in prtAlert, 40

Bursting
in prtOutput, 26

CapacityUnit
in prtInput, 21
in prtOutput, 25
CharacterSet
in prtGeneral, 20
CharacterSetIndex
in prtGeneral, 19
Class
in prtMarkerSupplies, 31
Color
in prtConsole, 38
ColorantID
in prtMarkerSupplies, 31
ConfigChanges
in prtGeneral, 19
ConsoleLocDescriptionalization
in prtConsole, 38
CounterUnit
in prtMarker, 30
Country
in prtGeneral, 19
CurrentJobCntlLangID
in prtChannel, 35
CurrentLevel
in prtInput, 21
CurrentLocatlization
in prtGeneral, 19

Decollating
in prtOutput, 27
DefaultCharSetIn
in prtInterpreter, 37
DefaultCharSetOut
in prtInterpreter, 37
DefaultID
in prtInput, 21
in prtInterpreter, 36
in prtMarker, 30
in prtMediaPath, 33
DefaultOrientation
in prtGeneral, 19
DefaultPageDescLangID
in prtChannel, 35
DefEastMarginaultID
in prtMarker, 31
Description
in prtAlert, 40
in prtGeneral, 19
in prtInput, 22
in prtInterpreter, 36
in prtMarkerSupplies, 31
in prtMediaPath, 33
in prtOutput, 26
DimUnit
in prtInput, 21
in prtOutput, 26

FeedAddressability
in prtInterpreter, 36

Group
in prtAlert, 40
GroupIndex
in prtAlert, 40

ID
in prtChannel, 35
in prtGeneral, 19
in prtInput, 21
in prtInterpreter, 36
in prtMarker, 30
in prtMarkerColorant, 31
in prtMarkerSupplies, 31
in prtMediaPath, 33
in prtOutput, 25
ifIndex
in prtChannel, 35
Index
in prtAlert, 39

JobCollated
in prtOutput, 27

LangFamily
in prtInterpreter, 36
LangLevel
in prtInterpreter, 36
Language
in prtGeneral, 19
LangVersion
in prtInterpreter, 36
Level
in prtMarkerSupplies, 31
LifeCount
in prtMarker, 30
LightID
in prtConsole, 38
LineText
in prtConsole, 38
Localization
in prtConsole, 38
Location
in prtAlert, 40
Lock
in prtConsole, 38

MarkerID
in prtMarkerColorant, 31
in prtMarkerSupplies, 31
MarkTech
in prtMarker, 30
MaxCapacity
in prtInput, 21
in prtMarkerSupplies, 31
in prtOutput, 25
MaxDimFeedDir
in prtOutput, 26
MaxDimXFeedDir
in prtOutput, 26
MaxMediaFeedDir
in prtMediaPath, 33
MaxMediaXFeedDir
in prtMediaPath, 33
MaxSpeed
in prtMediaPath, 33
MaxSpeedPrintUnit
in prtMediaPath, 33
MediaColor
in prtInput, 23
MediaDimFeedDir
in prtInput, 21
MediaDimXFeedDir
in prtInput, 21
MediaFormParts
in prtInput, 24
MediaName
in prtInput, 21
MediaType
in prtInput, 23
MediaWeight
in prtInput, 23
MinDimXFeedDir
in prtOutput, 26
MinMediaFeedDir
in prtMediaPath, 33
MinMediaXFeedDir
in prtMediaPath, 33

Model
in prtInput, 22
in prtOutput, 26

Name
in prtInput, 22
in prtOutput, 26
NorthMargin
in prtMarker, 30
NumberOfDisplayChars
in prtConsole, 38
NumberOfDisplayLines
in prtConsole, 38

OffsetStacking
in prtOutput, 27
OffTime
in prtConsole, 38
OnTime
in prtConsole, 38

PageCollated
in prtOutput, 27
PageDeliveryOrientation
in prtOutput, 26
PowerOnCount
in prtMarker, 30
ProcessColorants
in prtMarker, 30
prtAlertAlertCode, 40
prtAlertDescription, 40
prtAlertGroup, 40
prtAlertGroupIndex, 40
prtAlertIndex, 39
prtAlertLocation, 40
prtAlertSeverityLevel, 39
prtAlertTrainingLevel, 40
prtChannelCurrentJobCntlLangID, 35
prtChannelDefaultPageDescLangID , 35
prtChannelID, 35
prtChannelifIndex, 35
prtChannelState, 35
prtChannelStatus, 35
prtChannelType, 35
prtConsoleColor, 38
prtConsoleDescription, 38
prtConsoleLightID, 38
prtConsoleLineText, 38
prtConsoleLocalization, 38
prtConsoleLock, 38
prtConsoleNumberOfDisplayChars, 38
prtConsoleNumberOfDisplayLines, 38
prtConsoleOffTime, 38
prtConsoleOnTime, 38
prtGeneralCharacterSet, 20
prtGeneralCharacterSetIndex, 19
prtGeneralConfigChanges, 19
prtGeneralCountry, 19
prtGeneralCurrentLocalization, 19
prtGeneralDefaultOrientation, 19
prtGeneralDescription, 19
prtGeneralID, 19, 21
prtGeneralLanguage, 19
prtGeneralReset, 20
prtGeneralStatus, 19
prtInputCapacityUnit, 21
prtInputCurrentLevel, 21
prtInputDefaultID, 21
prtInputDescription, 22
prtInputDimUnit, 21
prtInputMaxCapacity, 21
prtInputMediaColor, 23
prtInputMediaDimFeedDir, 21
prtInputMediaDimXFeedDir, 21
prtInputMediaFormParts, 24
prtInputMediaName, 21
prtInputMediaType, 23
prtInputMediaWeight, 23
prtInputModel, 22
prtInputName, 22
prtInputSecurity, 22
prtInputSerialNumber, 22
prtInputStatus, 21
prtInputType, 21
prtInputVendorName, 22
prtInputVersion, 22
prtInterpreterDefaultCharSetIn, 37
prtInterpreterDefaultCharSetOut, 37
prtInterpreterDefaultID, 36
prtInterpreterDescription, 36
prtInterpreterFeedAddressability, 36
prtInterpreterID, 36
prtInterpreterLangFamily, 36
prtInterpreterLangLevel, 36
prtInterpreterLangVersion, 36
prtInterpreterTwoWay, 37
prtInterpreterVersion, 36
prtInterpreterXFeedAddressability, 37
prtMarkerAddressabilityFeedDir, 30
prtMarkerAddressabilityUnit, 30
prtMarkerAddressabilityXFeedDir, 30
prtMarkerColorantID, 31
prtMarkerColorantMarkerID, 31
prtMarkerColorantRole, 31
prtMarkerColorantTonality, 32
prtMarkerColorantValue, 32
prtMarkerCounterUnit, 30
prtMarkerDefaultID, 30
prtMarkerEastMargin, 31
prtMarkerID, 30
prtMarkerLifeCount, 30
prtMarkerMarkTech, 30
prtMarkerNorthMargin, 30
prtMarkerPowerOnCount, 30
prtMarkerProcessColorants, 30
prtMarkerSouthMargin, 31
prtMarkerSpotColorants, 30
prtMarkerStatus, 31
prtMarkerSuppliesClass, 31
prtMarkerSuppliesColorantID, 31
prtMarkerSuppliesDescription, 31
prtMarkerSuppliesID, 31
prtMarkerSuppliesLevel, 31
prtMarkerSuppliesMarkerID, 31
prtMarkerSuppliesMaxCapacity, 31
prtMarkerSuppliesSupplyUnit, 31
prtMarkerSuppliesType, 31
prtMarkerWestMargin, 31
prtMediaPathDefaultID, 33
prtMediaPathDescription, 33
prtMediaPathID, 33
prtMediaPathMaxMediaFeedDir, 33
prtMediaPathMaxMediaXFeedDir, 33
prtMediaPathMaxSpeed, 33
prtMediaPathMaxSpeedPrintUnit, 33
prtMediaPathMaxSpeedTimeUnit, 33
prtMediaPathMediaSizeUnit, 33
prtMediaPathMinMediaFeedDir, 33
prtMediaPathMinMediaXFeedDir, 33
prtMediaPathStatus, 33
prtMediaPathType, 33
prtOutputBursting, 26
prtOutputCapacityUnit, 25
prtOutputDecollating, 27
prtOutputDescription, 26
prtOutputDimUnit, 26
prtOutputID, 25
prtOutputJobCollated, 27
prtOutputMaxCapacity, 25
prtOutputMaxDimFeedDir, 26

prtOutputMaxDimXFeedDir, 26
prtOutputMinDimFeedDir, 26
prtOutputMinDimXFeedDir, 26
prtOutputModel, 26
prtOutputName, 26
prtOutputOffsetStacking, 27
prtOutputPageCollated, 27
prtOutputPageDeliveryOrientation, 26
prtOutputRemainingCapacity, 25
prtOutputSecurity, 26
prtOutputSerialNumber, 26
prtOutputSlipsheet, 27
prtOutputStackingOrder, 26
prtOutputStatus, 25
prtOutputType, 25
prtOutputVendorName, 26
prtOutputVersion, 26

RemainingCapacity
in prtOutput, 25
Reset
in prtGeneral, 20
Role
in prtMarkerColorant, 31

Security
in prtInput, 22
in prtOutput, 26
SerialNumber
in prtInput, 22
in prtOutput, 26
SeverityLevel
in prtAlert, 39
Slipsheet
in prtOutput, 27
SouthMargin
in prtMarker, 31
SpotColorants
in prtMarker, 30
StackingOrder
in prtOutput, 26
State
in prtChannel, 35
Status
in prtChannel, 35
in prtGeneral, 19
in prtInput, 21
in prtMarker, 31
in prtMediaPath, 33
in prtOutput, 25
SupplyUnit
in prtMarkerSupplies, 31

Tonality
in prtMarkerColorant, 32
TrainingLevel
in prtAlert, 40
TwoWay
in prtInterpreter, 37
Type
in prtChannel, 35
in prtInput, 21
in prtMarkerSupplies, 31
in prtMediaPath, 33
in prtOutput, 25

Value
in prtMarkerColorant, 32
VendorName
in prtInput, 22
in prtOutput, 26
Version
in prtInput, 22
in prtInterpreter, 36
in prtOutput, 26

WestMargin
in prtMarker, 31

XFeedAddressability
in prtInterpreter, 37