# Printer Management Interface

# Common MIB/MIF Specification

## Version 0.3, June 30, 1994

Files: prtspec3.dcr, prtspec3.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) 396-4515



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 3**
1.1. Network Printing Environment ....3
1.2. Printer Device Overview ....4
1.3. Categories of Printer Information ....4
**2. PRINTER MODEL 5**
2.1. Overview of the Printer Model ....6
2.2. Printer Sub-Units ....6
2.3. Read-Write Objects/Attributes ....12
2.4. Notational Conventions ....13
2.5. Common Enumerations ....15
**3. OBJECTS/ATTRIBUTES FROM OTHER MIB/MIF SPECIFICATIONS 17**
3.1. System Group objects ....17
3.2. System Controller ....17
3.3. Interface sGroup objects ....17
**4. GENERAL PRINTER GROUP 18**
4.1. Cover Table - Mandatory ....18
4.2. Localization Table - Mandatory ....18
4.3. CharacterSetTable - Mandatory ....19
4.4. General Printer - Extended - Optional ....19
**5. INPUT GROUP - MANDATORY 20**
5.1. Input Table - Mandatory ....20
5.2. Input Group Type Extended - Optional ....22
**6. MEDIA GROUP - OPTIONAL 23**
**7. OUTPUT GROUP - MANDATORY 25**
7.1. Output Table - Mandatory ....25
7.2. Output Group - Type Extended Objects - Optional ....26
7.3. Output - Dimensions Extended - Optional ....26
7.4. Output - Features - Optional ....27
**8. MARKER GROUP - MANDATORY 28**
8.1. Marker Table - Mandatory ....28
8.2. Marker Supplies Table - Mandatory ....29
8.3. Marker Colorant Table - Optional ....30
**9. MEDIA PATH GROUP - MANDATORY 32**
9.1. Media Path Table - Mandatory ....32
**10. CHANNEL GROUP - MANDATORY 34**
10.1. The channel table and its underlying structure ....34
10.2. Channel Table - Mandatory ....34
10.3. ChannelType Enumeration (enum type 2) ....35
**11. INTERPRETER GROUP - MANDATORY 37**
11.1. Interpreter Table - Mandatory ....37
11.2. IntpLangFam - Interpreter Language Family (enum type 2) ....38
**12. CONSOLE GROUP - MANDATORY 41**
12.1. Display Buffer Table - Mandatory ....41
**13. ALERTS GROUP - MANDATORY 43**
13.1. Alerts Table - Mandatory ....43
13.2. Alert Codes ....45
**APPENDIX A - GLOSSARY OF TERMS 46**

**APPENDIX B - MEDIA SIZE NAMES 48**
**APPENDIX C -MEDIA NAMES 50**
**APPENDIX D - ROLES OF USERS 52**
Background....52
Proposed Print System Requirements Areas ....52
Proposed Role Models....53
Matrix of Requirement Areas and Role Models ....53
**APPENDIX E - DRAFT SPECIFICATION FOR THE FINISHING SUB-UNIT 55**
**FINISHER GROUP - OPTIONAL 56**
Output Associated Finishers Table - Optional ....56
Finishing Configuration - Optional ....57
Finisher Table - Optional ....57
Finisher Enumerations....57
**APPENDIX F - PARTICIPANTS 59**
**APPENDIX Y - ISSUES 60**
Changes from prtspec Version 0.1, 15-May to Version 0.2, 31-May ....71
Changes from Version 0.7, 27-Apr to Version 0.1, 15-May....72
Changes from Version 0.6, 26-Apr, to Version 0.7, 27-Apr....73
Changes from Version 0.2, 25-Apr, to Version 0.6, 26-Apr....74
Changes from Version 0.1, 23-Apr, to Version 0.2, 25-Apr....74
Changes to create Version 0.1, 23-Apr....74
**INDEX 76**

# 1. Introduction

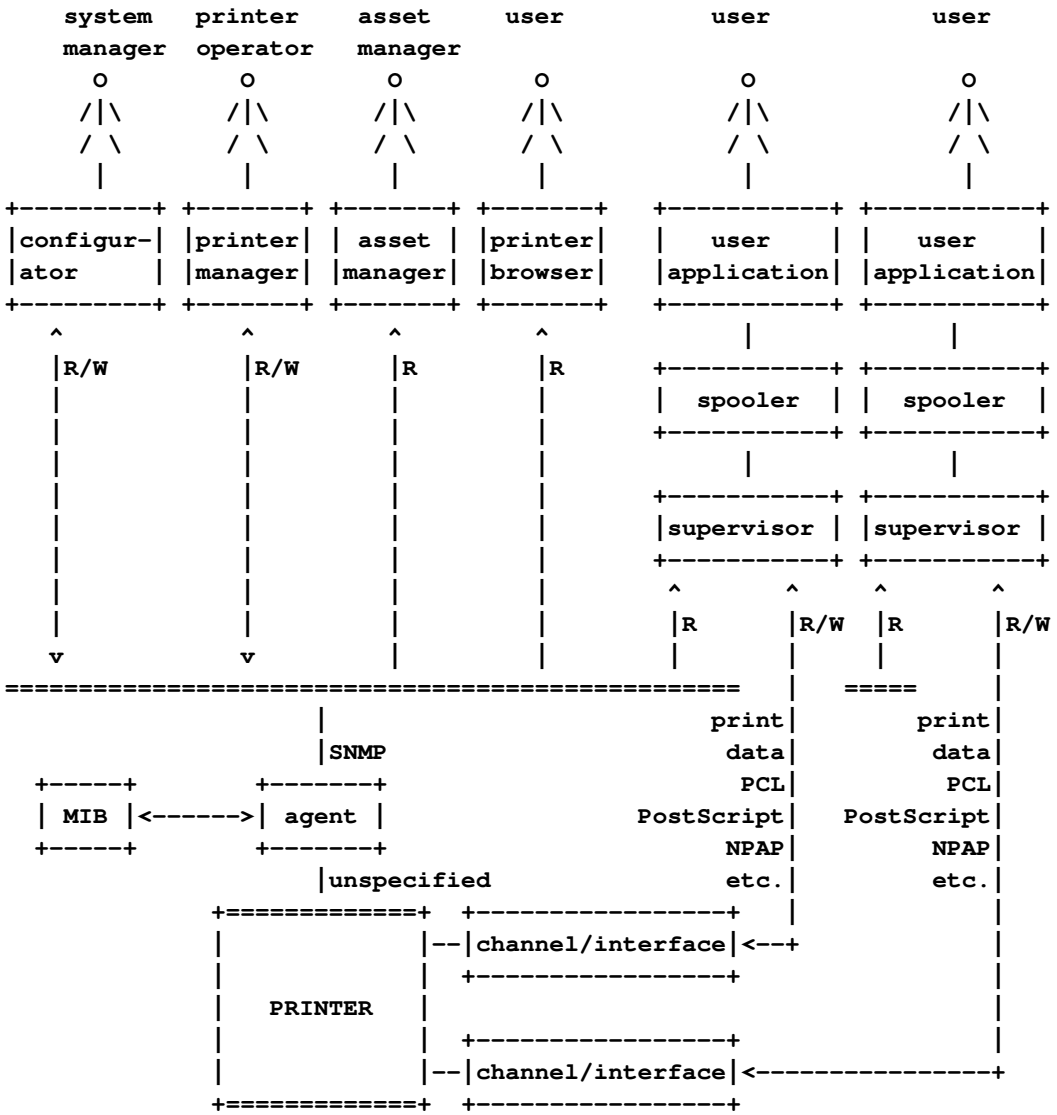
The Printer Management Interface (PMI) Common MIB/MIF Specification specifies the semantics of the IETF Printer MIB and the DMTF Printer MIF in a manner that is independent of the syntax for MIB and MIF specifications. After this document is relatively stable, it will be converted into the appropriate 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.

All trademarks and registered trademarks are the property of their respective owners.

## 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.

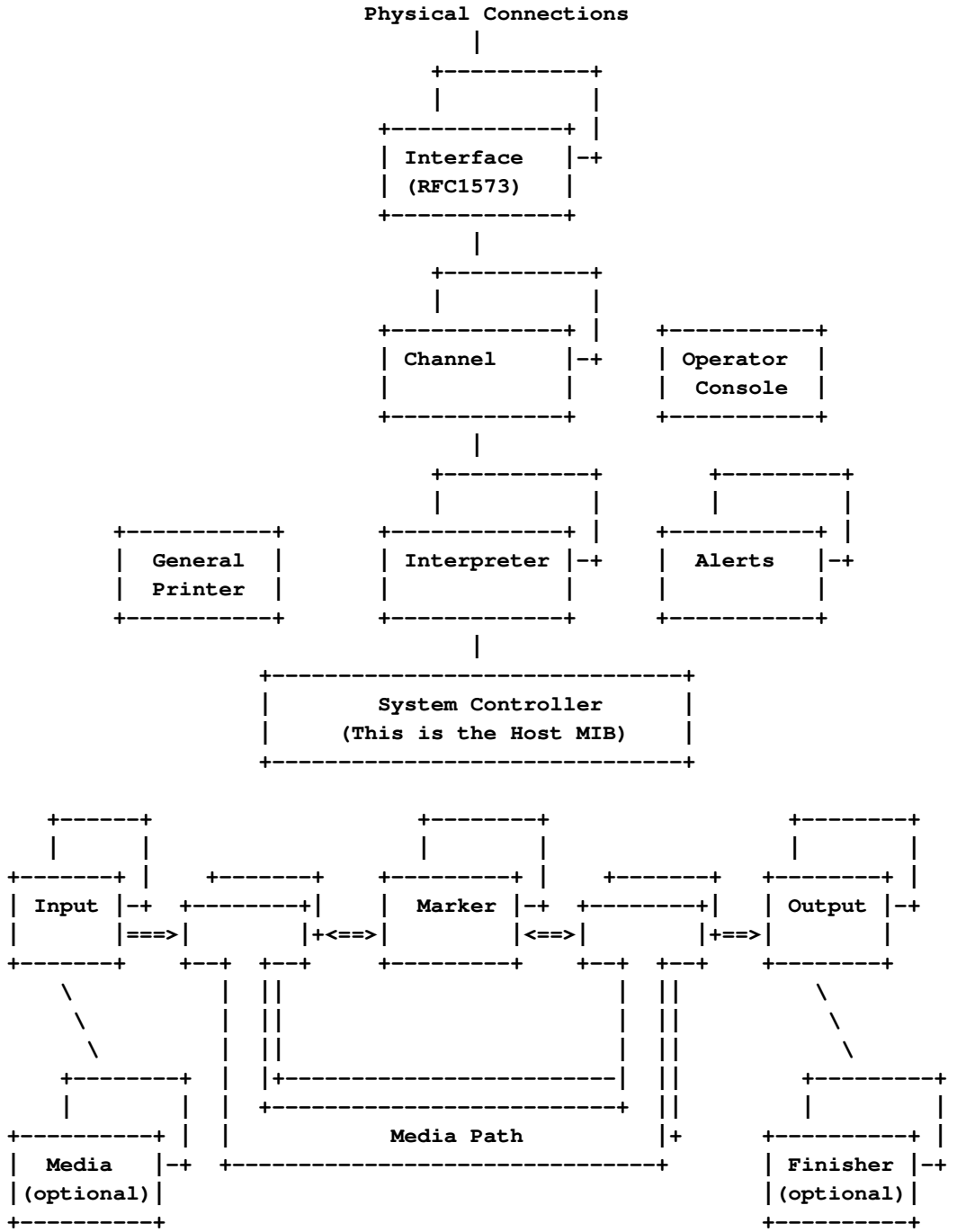
Description, status and alert information about printer can be thought of as a data base describing the printer. The management application for a printer will want to view the printer data base differently depending on how and for what purposes the information in the data base is needed.

## 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 13 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 13 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 13 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 enumeration is responsible for identifying the possible character sets that are used by the interpreters, the operator console and management requests for display objects. 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 the 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 sub-unit and each finishing sub-unit may be associated with one or more output sub-units.

The exact interaction and sequencing between an output sub-unit and associated finishers is not specified by the model. Nor does this standard define the management of the finisher sub-unit. Both the management of finishers and their interaction with the Output sub-units is the subject of another working group.

Finishers are not described in this document.

### 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 sub-unit 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 sub-unit, 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.

#### 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.

#### 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 sub-unit is unavailable for use (this is the same as "broken" or "down" in other terminology's)

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

**Moving on- or off-line** - The sub-unit 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 do so 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.

### 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 }
```

### 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 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).

### 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.

#### 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.

The agent needs to have an algorithm implemented for the management of the alert table, especially in the face of combinations of binary and simple alerts that would overflow the storage capacity of the table. When the table is full and a new alert needs to be added, an old alert needs to be deleted. The alert to be deleted should be chosen using the following rules

1. Find a non-critical simple alert and delete it. If there are multiple simple alerts, it is suggested that the oldest one be chosen. If there are no non-critical simple alerts, then,
2. Find a non-critical binary alert and delete it. If there are multiple non-critical binary alerts, it is suggested that the oldest one be chosen. If there are no non-critical binary alerts, then,
3. Find a critical (binary) alert and delete it. If there are multiple non-critical binary alerts, it is suggested that the oldest one be chosen. Agent implementors are encouraged to provide at least enough storage space for the maximum number of critical alerts that could occur simultaneously. Note that all critcial alerts are binary.

Note that because the Alert Index is a monotonically increasing integer, there will be gaps in the values in the table when an alert is deleted. Such gaps can be detected by the management application and used 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 a printer implementation might only permit a write, if the printer could not sense the value itself. Therefore, the ability to change the value of a Read/Write object/attribute may depend on the imlementation of the agent. Note that even though some objects/attributes explicitely state the behaviour of conditional ability to change, any Read/Write object/attribute may act that way.

Generally, 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 ahve a corresponding MIF that specifies identical naming, syntax, and semantics so that a product described by the MIF can be managed using SNMP and the Printer MIB via a trivial translation.

### 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.

**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**, **prtInputDefaultInputIndex**, **prtInputIndex, and prtMediaPathMaxSpeed**. These are the names for objects that are most accessable to simple management applications that may just display object-name/value pairs; that is, the OID for an object is displayed with the name that was used to define that OID in the MIB.

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.

#### DMTF MIF Attribute Names

DMTF allows attribute names to be quoted strings containing any ISO Latin-1 character, including SPACE. So DMTF names could be different from SNMP names and include spaces and special characters. However, the advantages of names with spaces was not seen to outweigh the problem of managing two sets of names. Therefore, the same names will be used both for SNMP and DMTF.

### 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 **SubUnitStatus**.

### 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 common enumerations are listed at the end of this section; enumerations used with only one object/attribute are listed with that object/attribute.

#### 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 **ch**, respectively are used. Examples of the Printer MIB enumerations for the **SubUnitStatus** and **IntpLangFam** data types, respectively, are: **bothCriticalAndNonCritical**, and **langPCL, langPS** and **langISO6429**.

#### 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 use the same enum symbols as are used in the SNMP MIB.

#### 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.

## 2.5. Common Enumerations

The common enumerations are listed in this section.

### 2.5.1. Sub-Unit Status

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

| | | |
|---|---|---|
| SubUnitStatus (enum type 1) (used > 1) | | |
| 1 | Other | |
| 2 | Unknown | |
| 3 | Available | |
| 4 | Critical | -- unavailable, in critical aleter state |
| 5 | Noncritical | -- available, in non-critical alert state |
| 6 | Critical And Noncritical | -- unavailable, in both critical and non-critical aleter state |
| 7 | On Request | |
| 8 | Unavailable | |
| 9 | Busy Temporally Unavailable | |

### 2.5.2. 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 (both here and in the descriptions) have been harmonized so that the same numeric value is used for the same unit in all enumerations.

| AddrUnit (enum type 1) (used >1) | | MediaUnit (enum type 1) (used > 1) | | CapacityUnit (enum type 1) (used >1) | |
|---|---|---|---|---|---|
| 1 | (reserved) | 1 | (reserved) | 1 | (reserved) |
| 2 | (reserved) | 2 | (reserved) | 2 | (reserved) |
| 3 | Ten Thousandths Of Inches | 3 | Ten Thousandths Of Inches | 3 | Ten Thousandths Of Inches |
| 4 | Micrometers | 4 | Micrometers | 4 | Micrometers |
| | | | | 8 | Sheets |
| | | | | 16 | Feet |
| | | | | 17 | Meters |

### 2.5.3. Miscellaneous

| PresentOnOff (enum type 1) (used > 1) | | | |
|---|---|---|---|
| 1 | Other | | |
| 2 | (reserved) | | |
| 3 | On | | |
| 4 | Off | | |
| 5 | NotPresent | | |

Notes:

# 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

All objects in the System Group of MIB-II (RFC 1213) shall be implemented.

## 3.2. System Controller

The System Controller is represented by the Storage and Device Groups of the Host Resources MIB (RFC 1514). These are the only Groups that are required to be implemented. Other Groups (System, Running Software, Running Software Performance, and Installed Software) may be implemented at the discretion of the implementor.

## 3.3. Interface sGroup objects

All objects in the Interfaces Group of MIB-II (RFC 1213) shall be implemented.

# 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.<br><br>Management applications may cache infrequently changed configuration information about sub-units on the printer. This object should be incremented wheneer the agent wishes such applications to invalidate that cache and re-download all of this configuration information, thereby signalling a change of the printers configuration.<br><br>For example, if an input tray that contained paper of different dimensions was added, this counter would be incremented.<br><br>As an additional example, this counter would not be incremented when an input tray is removed or the level of an input device is changed. |
| CurrentLocalization | Read-Write | Integer32 | | The value of **prtGeneralLocalizationIndex** corresponding to the current language, country and character set to be used for **prt*Xxxx*Description** string values. If there no strings under the control of this object that must be localizaed the value shall be zero toherwise the value must be 1 or greater. |

Notes:

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

## 4.1. Cover Table - Mandatory

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

| **prtGeneral** | **Mandatory** | | **prtCoverTable** | |
|---|---|---|---|---|
| Object Name | **Access** | **Type** | **Value** | **Description** |
| CoverIndexCover | Read-Only | Integer32 | | A unique value used by the printer to identify this Cover sub-unit. Although these values may change due to a major reconfiguration of the device (e.g. the addition of new Cover sub-units to the printer), values are expected to remain stable across successive printer power cycles. Must be 1 or greater. |
| Description | Read-Only | String(255) | | The manufacturer provided Cover sub-mechanismname in the localization specified by **CurrentLocalization**. |
| Status | Read-Only | Enum (2) | CoverStatus | The status of this Cover sub-unit.<br>This value is a type 2 enum.<br>1 Other<br>2 (reserved)<br>3 Door Open<br>4 Door Closed<br>5 Interlock Open<br>6 Interlock Closed |

## 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** |

| | | | | |
|---|---|---|---|---|
| LocalizationIndexLocalization | Read-Only | Integer32 | | A unique value used by the printer to identify this Localization entry. Although these values may change due to a major reconfiguration of the device (e.g. the addition of new Cover sub-units to the printer), values are expected to remain stable across successive printer power cycles. Must be 1 or greater. |
| 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, ... |
| LocalizationCharacterSetIndex | Read-Only | Integer32 | | The value of **prtGeneralCharacterSetIndex** identifying the character set used for this localization. Must be 1 or greater. |

Notes:

1. Added **Localization** to CharacterSetIndex name, so that it is different than the next object
2. Added Index column to the Localization table with the name **LocalizationIndex.**

### 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 |
| CharacterSetIndexCharacterSet | Read-Only | Integer32 | | A unique value used by the printer to identify this CharacterSet entry. Although these values may change due to a major reconfiguration of the device (e.g. the addition of new CharacterSet s to the printer), values are expected to remain stable across successive printer power cycles. Must be 1 or greater. |
| CharacterSet | Read-Only | String(63) | | A character set name or identifier that identifies this character set. |

Notes:

1. Added Index column ot the Character Set Table with the name **CharacterSetIndex**.

### 4.4. General Printer - Extended - Optional

| **prtGeneral - Extended** | **Optional** | | | |
|---|---|---|---|---|
| Object Name | Access | Type | Value | Description |
| Reset | Read-Write | Enum (3) | ResetType | Writing a value greater than 3 causes a reset. Writing a value of 3 has no effect. Writing values 1 or 2 is not allowed.<br>This value is a type 3 enum.<br>3 NotReseting<br>4 Power Cycle Reset<br>5 Reset NVRAM<br>6 Reset To Factory Defaults |

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 |
| DefaultIndex | Read-Write | Integer32 | | The value of **prtInputIndex** corresponding tothe default input sub-unit; that is, this object selects the default source of input media. Must be 1 or greater |

Notes:

## 5.1. Input Table - Mandatory

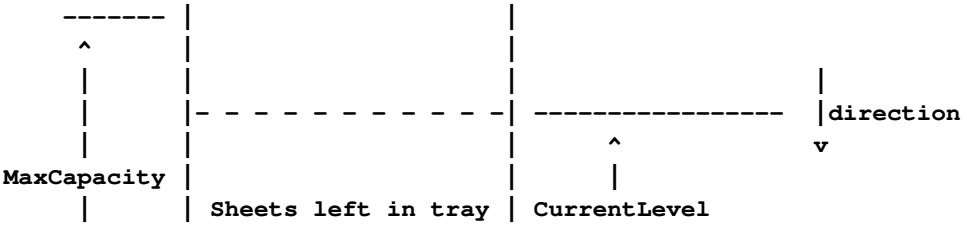
| **prtInput** | **Mandatory** | | **prtInputTable** | |
|---|---|---|---|---|
| Object Name | Access | Type | Value | Description |
| Index | Read-Only | Integer32 | | A unique value used by the printer to identify this input sub-unit. Although these values may change due to a major reconfiguration of the device (e.g. the addition of new input sub-units to the printer), values are expected to remain stable across successive printer power cycles. Must be 1 or greater |
| Type | Read-Only | Enum (2) | InputType | The type of technology (discriminated primarily according to feeder mechanism type) employed by the input sub-unit. Note, the Optional Input Class provides for a descriptor field to further qualify the other choice.<br>This value is a type 2 enum.<br>1 Other<br>2 Unknown<br>3 Sheet Feed Auto Removable Tray<br>4 Sheet Feed Auto Non Removable Tray<br>5 Sheet Feed Manual<br>6 Continuous Roll<br>7 Continuous Fan Fold |
| DimUnit | Read-Write | Enum (1) | MediaUnit | The unit of measurement for use in calculating and relaying dimensional values for this input sub-unit.<br>This value is a type 1 enum. |
| MediaDimFeedDirDeclared | Read-Write | Integer32 | | This object provides the value of the declared dimension, in the feed direction, of the media that is (or, if empty, was or will be) in this input sub-unit. The feed direction is the direction in which the media is feed on this sub-unit. This dimension is measured in input sub-unit dimensional units (DimUnit). If this input sub-unit can reliably sense this value, the value is sensed by the printer and is read-only; otherwise, the value may be written (by a Remote Contol Panel or a Management Application). The value (-1) means other and specifically indicates that this sub-unit places no restriction on this parameter. The value (-2) indicates unknown. |
| MediaDimXFeedDirDeclared | Read-Write | Integer32 | | This object provides the value of the declared dimension, in the cross feed direction, of the media that is (or, if empty, was or will be) in this input sub-unit. The cross feed direction is ninety degrees relative to the feed direction associated with this sub-unit. This dimension is measured in input sub-unit dimensional units (DimUnit). If this input sub-unit can reliably sense this value, the value is sensed by the printer and is read-only; otherwise, the value may be written (by a Remote Contol Panel or a Management Application). The value (-1) means other and specifically indicates that this sub-unit places no restriction on this parameter. The value (-2) indicates unknown. |

| | | | | |
|---|---|---|---|---|
| MediaDimFeedDirChosen | Read-Only | Integer32 | | The printer will act as if media of the chosen dimension (in the feed direction) is present in this input source. Note that this value will be used even if the input tray is empty. Feed dimension measurements are taken parallel relative to the feed direction associated with that sub-unit and are in input sub-unit dimensional units (DimUnit). If the printer supports the declared dimension, the granted dimension is the same as the declared dimension. If not, the granted dimension is set to the closest dimension that the printer supports when the declared dimension is set. The value (-1) means other and specifically indicates that this sub-unit places no restriction on this parameter. The value (-2) indicates unknown. |
| MediaDimXFeedDirChosen | Read-Only | Integer32 | | The printer will act as if media of the chosen dimension (in the cross feed direction) is present in this input source. Note that this value will be used even if the input tray is empty. The cross feed direction is ninety degrees relative to the feed direction associated with this sub-unit. This dimension is measured in input sub-unit dimensional units (DimUnit). If the printer supports the declared dimension, the granted dimension is the same as the declared dimension. If not, the granted dimension is set to the closest dimension that the printer supports when the declared dimension is set. The value (-1) means other and specifically indicates that this sub-unit 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 sub-unit.<br>This value is a type 1 enum. |
| MaxCapacity | Read-Write | Integer32 | | The maximum capacity of the input sub-unit in input sub-unit capacity units (CapacityUnit). There is no convention associated with the media itself so this value reflects claimed capacity. If this input sub-unit can reliably sense this value, the value is sensed by the printer and is read-only; otherwise, the value may be written (by a Remote Contol Panel or a Management Application). The value (-1) means other and specifically indicates that the sub-unit 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 sub-unit capacity units (CapacityUnit). If this input sub-unit can reliably sense this value, the value is sensed by the printer and is read-only; otherwise, the value may be written (by a Remote Contol Panel or a Management Application). The value (-1) means other and specifically indicates that the sub-unit places no restrictions on this parameter. The value (-2) means unknown. The value (-3) means that the printer knows that at least one unit remains. |
| Status | Read-Only | Enum (1) | SubUnitStatus | The current status of this input sub-unit<br>This value is a type 1 enum.. |
| MediaName | Read-Write | String(63) | | A description of the media contained in this input sub-unit; This description is intended for display to a human operator. This description is not processed by the printer. It is used to provide information not expressible in terms of the other media attributes (ChosenMediaDimFeedDir, ChosenMediaDimXFeedDir, MediaWeight, MediaType). An example would be "Hillary's Pink Presidential Letterhead". |

Notes:

*1. Descriptions of Declared and Grantd Media(X)Feed have been rewritten to descibed the roles of the Declared and Granted values and to indicate when the declared value can be written.*

*2. Descriptions of Max Capacity and Current Level have been modified to indicate when the values can be written and to match the style of the objects in Note 1.*



## 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 sub-unit. |
| VendorName | Read-Only | String(63) | | The vendor name of this input sub-unit. |
| Model | Read-Only | String(63) | | The model name of this input sub-unit. |
| Version | Read-Only | String(63) | | The version of this input sub-unit. |
| SerialNumber | Read-Only | String(63) | | The serial number assigned to this input sub-unit. |
| Description | Read-Only | String(255) | | A free-form text description of this input sub-unit in the localization specified by **prtGeneralCurrentLocalization**. |
| Security | Read-Write | Enum (1) | PresentOnOff | Indicates if this input sub-unit 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 sub-unit. The value (-2) means unknown. |
| MediaType | Read-Write | String(63) | | The name of the type of medium associated with this input sub-unit. The standardized string values from ISO 10175 (DPA) and ISO 10180 (SPDL) are:<br>**stationery** Separately cut sheets of an opaque material<br>**transparency** Separately cut sheets of a transparent material<br>**envelope** Envelopes that can be used for conventional mailing purposes<br>**envelope-plain** Envelopes that are not preprinted and have no windows<br>**envelope-window** Envelopes that have windows for addressing purposes<br>**continuous-long** Continuously connected sheets of an opaque material connected along the long edge<br>**continuous-short** Continuously connected sheets of an opaque material connected along the short edge<br>**tab-stock** Media with tabs<br>**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<br>**labels** Label stock<br>**multi-layer** Form medium composed of multiple layers which are pre-attached to one another; e.g., for use with impact printers<br>Implementors may add additional string values. The naming conventions in ISO 9070 are recommended in order to avoid potential name clashes. |
| MediaColor | Read-Write | String(63) | | The name of the color of the medium associated with this input sub-unit. The standardized string values from ISO 10175 (DPA) and ISO 10180 (SPDL) are:<br>**other**<br>**unknown**<br>**white**<br>**pink**<br>**yellow**<br>**buff**<br>**goldenrod**<br>**blue**<br>**green**<br>**transparent**<br>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 sub-unit if the medium is a multi-part form. The value (-1) means other and indicates the medium does not have a fixed number of form parts. The value (-2) means unknown. |

Notes:

# 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 |
| DefaultIndex | Read-Write | Integer32 | | The value of **prtOutputIndex** corresponding tothe default output sub-unit; that is, this object selects the default output destination. Must be 1 or greater. |

Notes:

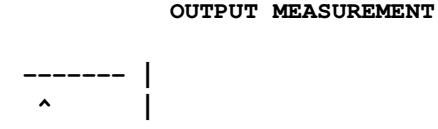
*1. Changed the descripton to match the style of Index references*

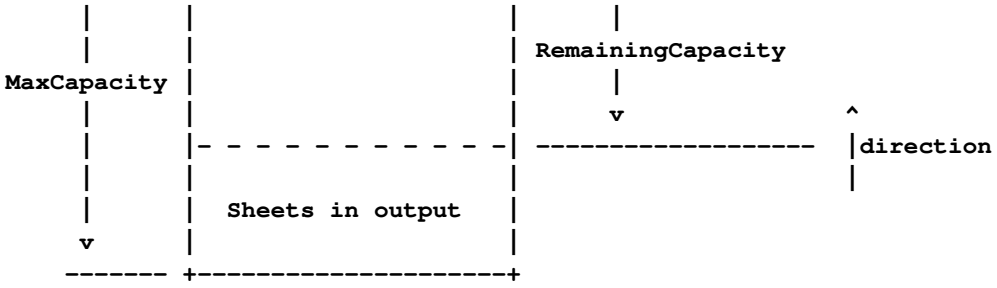
## 7.1. Output Table - Mandatory

| **prtOutput** | **Mandatory** | | **prtOutputTable** | |
|---|---|---|---|---|
| Object Name | Access | Type | Value | Description |
| Index | Read-Only | Integer32 | | A unique value used by this printer to identify this output sub-unit. Although these values may change due to a major reconfiguration of the sub-unit (e.g. the addition of new output sub-units to the printer), values are expected to remain stable across successive printer power cycles. Must be 1 or greater. |
| Type | Read-Only | Enum (2) | OutputType | The type of technology supported by this output sub-unit.<br>This value is a type 2 enum.<br>1 Other<br>2 Unknown<br>3 Removable Bin<br>4 Unremovable Bin<br>5 Continuous Roll Device<br>6 Mail Box |
| CapacityUnit | Read-Write | Enum (1) | CapacityUnit | The unit of measurement for use in calculating and relaying capacity values for this output sub-unit.<br>This value is a type 1 enum. |
| MaxCapacity | Read-Write | Integer32 | | The maximum capacity of this output sub-unit in output sub-unit capacity units (CapacityUnit). There is no convention associated with the media itself so this value essentially reflects claimed capacity. If this output sub-unit can reliably sense this value, the value is sensed by the printer and is read-only; otherwise, the value may be written (by a Remote Contol Panel or a Management Application). The value (-1) means other and specifically indicates that the sub-unit 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 sub-unit capacity units (CapacityUnit) of this output sub-unit. If this output sub-unit can reliably sense this value, the value is sensed by the printer and is read-only; otherwise, the value may be written (by a Remote Contol Panel or a Management Application). The value (-1) means other and specifically indicates that the sub-unit places no restrictions on this parameter. The value (-2) means unknown.The value (-3) means that the printer knows that there remains capacity for at least one unit. |
| Status | Read-Only | Enum (1) | SubUnitStatus | The current status of this output sub-unit.<br>This value is a type 1 enum. |

Notes:

*1. Descriptions of Max Capacity and RemainingCapacity have been modified to indicate when the values can be written .*






## 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 sub-unit. |
| VendorName | Read-Only | String(63) | | The vendor name of this output sub-unit. |
| Model | Read-Only | String(63) | | The name assigned to this output sub-unit. |
| Version | Read-Only | String(63) | | The version of this output sub-unit. |
| SerialNumber | Read-Only | String(63) | | The serial number assigned to this output sub-unit. |
| Description | Read-Only | String(255) | | A free-form text description of this output sub-unit in the localization specified by **prtGeneralCurrentLocalization**. |
| Security | Read-Write | Enum (1) | PresentOnOff | Indicates if this output sub-unit has some security associated with it. and if that security is enabled or not. |

Notes:

## 7.3. Output - Dimensions Extended - Optional

| prtOutput - DimExtended | Optional | | | prtOutputTableAugment |
|---|---|---|---|---|
| Object Name | Access | Type | Value | Description |
| DimUnit | Read-Write | Enum (1) | MediaUnit | The unit of measurement for use in calculating and relaying dimensional values for this output sub-unit.<br>This value is a type 1 enum. |
| MaxDimFeedDir | Read-Write | Integer32 | | The maximum dimensions supported by this output sub-unit for measurements taken parallel relative to the feed direction associated with that sub-unit in output sub-unit dimensional units (DimUnit). If this output sub-unit can reliably sense this value, the value is sensed by the printer and is read-only; otherwise, the value may be written (by a Remote Contol Panel or a Management Application). |
| MaxDimXFeedDir | Read-Write | Integer32 | | The maximum dimensions supported by this output sub-unit for measurements taken ninety degrees relative to the feed direction associated with that sub-unit in output sub-unit dimensional units (DimUnit). If this output sub-unit can reliably sense this value, the value is sensed by the printer and is read-only; otherwise, the value may be written (by a Remote Contol Panel or a Management Application). |
| MinDimFeedDir | Read-Write | Integer32 | | The minimum dimensions supported by this output sub-unit for measurements taken parallel relative to the feed direction associated with that sub-unit in output sub-unit dimensional units (DimUnit). If this output sub-unit can reliably sense this value, the value is sensed by the printer and is read-only; otherwise, the value may be written (by a Remote Contol Panel or a Management Application). |
| MinDimXFeedDir | Read-Write | Integer32 | | The minimum dimensions supported by this output sub-unit for measurements taken ninety degrees relative to the feed direction associated with that sub-unit in output sub-unit dimensional units (DimUnit). If this output sub-unit can reliably sense this value, the value is sensed by the printer and is read-only; otherwise, the value may be written (by a Remote Contol Panel or a Management Application). |

Notes:

*1. Descriptions of Max and Min Dimension(X)Feed have been modified to indicate when the values can be written*

## 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 sub-unit. “FirstToLast” means that as pages are output the front of the next page is placed against the back of the previous page. “LasttoFirst means that as pages are output the back of the next page is placed against the front of the previous page.<br>This value is a type 1 enum.<br>1 (reserved)<br>2 Unknown<br>3 First To Last<br>4 Last To First |
| PageDeliveryOrientation | Read-Write | Enum (1) | PageDeliveryOrientation | The reading surface that will be "up" when pages are delivered to the associated output sub-unit. 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).<br>This value is a type 1 enum.<br>1 (reserved)<br>2 (reserved)<br>3 Face Up<br>4 Face Down |
| Bursting | Read-Write | Enum (1) | PresentOnOff | This object indicates that the output sub-unit 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.<br>This value is a type 1 enum. |
| Decollating | Read-Write | Enum (1) | PresentOnOff | This object indicates that the output sub-unit 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.<br>This value is a type 1 enum. |
| PageCollated | Read-Write | Enum (1) | PresentOnOff | This object indicates that the output sub-unit supports page collation, and if so, whether the feature is enabled.<br>This value is a type 1 enum. |
| OffsetStacking | Read-Write | Enum (1) | PresentOnOff | This object indicates that the output sub-unit supports offset stacking, and if so, whether the feature is enabled.<br>This value is a type 1 enum. |
| Slipsheet | Read-Write | Enum (1) | PresentOnOff | This object indicates that the output sub-unit supports slipsheet insertion between sets, and if so, whether the feature is enabled.<br>This value is a type 1 enum. |

Notes:

*1. The stacking order enumeratin values hasvebeen renamed to "FirstToLast" and "LastToFirst" and thedescription has been changed to avoid any reference to how the input arrived at the printer. The description is now expressed solely in terms of how output pages are processed.*

*2.. Should Slipsheet be in this group or should it be in the Finisher Group because it involves supplies.*

# 8. 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 sub-unit can have its own set of characteristics associated with it, such as marking technology and resolution.

| **prtMarker** | **Mandatory** | | | |
|---|---|---|---|---|
| Object Name | Access | Type | Value | Description |
| DefaultIndex | Read-Write | Integer32 | | The value of **prtMarkerIndex** corresponding tothe default markersub-unit; that is, this object selects the default marker. Must be 1 or greater. |

Notes:

## 8.1. 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 |
| Index | Read-Only | Integer32 | | A unique value used by the printer to identify this marking sub-unit. Although these values may change due to a major reconfiguration of the device (e.g. the addition of new marking sub-units to the printer), values are expected to remain stable across successive printer power cycles. Must be 1 or greater. |
| MarkTech | Read-Only | Enum (2) | MarkingTech | The type of marking technology used for this marking sub-unit. This value is a type 2 enum.<br>1 Other<br>2 (reserved)<br>3 Electrophotographic LED<br>4 Electrophotographic Laser<br>5 Electrophotographic Other<br>6 Impact Moving Head Dot Matrix 9pin<br>7 Impact Moving Head Dot Matrix 24pin<br>8 Impact Moving Head Dot Matrix Other<br>9 Impact Moving Head Fully Formed<br>10 Impact Band<br>11 Impact Other<br>12 Inkjet Aqueous<br>13 Inkjet Solid<br>14 Inkjet Other<br>15 Pen<br>16 Thermal Transfer<br>17 Thermal Sensitive<br>18 Thermal Diffusion<br>19 Thermal Other<br>20 Electroerosion<br>21 Electrostatic<br>22 Photographic Microfiche<br>23 Photographic Imagesetter<br>24 Photographic Other<br>25 Ion Deposition<br>26 E beam<br>27 Typesetter |

| | | | | |
|---|---|---|---|---|
| CounterUnit | Read-Only | Enum (1) | CounterUnit | The unit that will be used by the printer when reporting counter values for this marking sub-unit. 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.<br>This value is a type 1 enum.<br>1 (reserved)<br>2 (reserved)<br>3 Ten Thousandths Of Inches -- .0001<br>4 Micrometers<br>5 Characters<br>6 Lines<br>7 Impressions<br>8 Sheets<br>9 Dot Row<br>11 Hours<br>16 Feet<br>17 Meters |
| 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. Must be 0 or greater |
| SpotColorants | Read-Only | Integer32 | | The number of spot colors supported by this marker. The value of this object and ProcessColorants cannot both be 0. Must be 0 or greater. |
| AddressabilityUnit | Read-Only | Enum (1) | AddrUnit | The unit of measure of distances<br>This value is a type 1 enum. |
| AddressabilityFeedDir | Read-Only | Integer32 | | The number of addressable marking positions in the feed direction per 10,000 ;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 number of addressable marking positions in the cross feed direction in 10,000 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 , in units identified by AddressabilityUnit., from the leading edge of the medium as the medium flows throught the marking engine with the side to be imaged facing the observer. The leading edge is the North edge and the other edges are defined by the normal compass layout of directions with the compass facing the observer. Printing within the area bounded by all four margins is guaranteed for all interpreters. The value (-2) means unknown. |
| SouthMargin | Read-Only | Integer32 | | The margin from the South edge (see NorthMargin) of the medium in units identified by AddressabilityUnit. Printing within the area bounded by all four margins is guaranteed for all interpreters. The value (-2) means unknown. |
| WestMargin | Read-Only | Integer32 | | The margin from the West edge (see NorthMargin) of the medium in units identified by AddressabilityUnit. Printing within the area bounded by all four margins is guaranteed for all interpreters. The value (-2) means unknown. |
| EastMargin | Read-Only | Integer32 | | The margin from the East edge (see NorthMargin) of the medium in units identified by AddressabilityUnit. Printing within the area bounded by all four margins is guaranteed for all interpreters. The value (-2) means unknown. |
| Status | Read-Only | Enum (1) | SubUnitStatus | The current status of this marker sub-unit.<br>This value is a type 1 enum. |

Notes:

*1. The descriptions of Addressability and Margins have been changed to bring them in line with the Working Group minutes.*

## 8.2. Marker Supplies Table - Mandatory

| **prtMarkerSupplies** | **Mandatory** | | **prtMarkerSuppliesTable** | |
|---|---|---|---|---|
| Object Name | Access | Type | Value | Description |

| | | | | |
|---|---|---|---|---|
| Index | 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. Must be 1 or greater. |
| MarkerIndex | Read-Only | Integer32 | | The value of **prt MarkerIndex** corresponding tothe marking sub-unit with which this marker supply sub-unit is associated. Must be 1 or greater. |
| ColorantIndex | Read-Only | Integer32 | | The value of **prt MarkerColorantIndex** corresponding to the colorant with which this marker supply entity is associated. This value shall be 0 if there is no colorant table. Must be 0 or greater. |
| Class | Read-Only | Enum (1) | MkrSupplyClass | Indicates whether this supply entity represents a supply container that is consumed or a receptacle that is filled.<br>This value is a type 1 enum.<br>1 Other<br>2 (reserved)<br>3 Supply That Is Consumed<br>4 Receptacle That Is Filled |
| Type | Read-Only | Enum (2) | MkrSupplyType | The type of this supply.<br>This value is a type 2 enum.<br>1 Other<br>2 Unknown<br>3 Toner<br>4 Waste Toner<br>5 Ink<br>6 Ink Cartridge<br>7 Ink Ribbon<br>8 Waste Ink<br>9 OPC<br>10 Developer<br>11 Fuser Oil<br>12 Solid Wax<br>13 Ribbon Wax<br>14 Waste Wax |
| Description | Read-Only | String(255) | | The description of this supply container/receptacle in the localization specified by **prtGeneralCurrentLocalization**. |
| SupplyUnit | Read-Only | Enum (1) | MkrSupplyUnit | Unit of this marker supply container/receptacle.<br>This value is a type 1 enum.<br>1 (reserved)<br>2 (reserved)<br>3 Ten Thousandths Of Inches<br>4 Micrometers<br>8 Sheets<br>12 Thousandths Of Ounces<br>13 Tenths Of Grams<br>14 Hundredths Of Fluid Ounces<br>15 Tenths Of Milliliters |
| MaxCapacity | Read-Write | Integer32 | | The maximum capacity of this supply container/receptacle expressed in SupplyUnit. If this supply container/receptacle can reliably sense this value, the value is sensed by the printer and is read-only; otherwise, the value may be written (by a Remote Contol Panel or a Management Application). The value (-1) means other and specifically indicates that the sub-unit places no restrictions on this parameter. The value (-2) means unknown. |
| Level | Read-Write | Integer32 | | The current level if this supply is a container; the remaining space if this supply is a receptacle. If this supply container/receptacle can reliably sense this value, the value is sensed by the printer and is read-only; otherwise, the value may be written (by a Remote Contol Panel or a Management Application). The value (-1) means other and specifically indicates that the sub-unit places no restrictions on this parameter. The value (-2) means unknown.The value (-3) means that the printer knows thatthere is some supply/remaining space, respectively. |

Notes

## 8.3. Marker Colorant Table - Optional

| **prtMarkerColorant** | **Optional** | | **prtMarkerColorantTable** | |
|---|---|---|---|---|
| Object Name | Access | Type | Value | Description |

| | | | | |
|---|---|---|---|---|
| Index | 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. Must be 1 or greater. |
| MarkerIndex | Read-Only | Integer32 | | The value of **prt MarkerIndex** corresponding tothe marker sub-unit with which this colorant entry is associated. Must be 1 or greater. |
| Role | Read-Only | Enum (1) | ColorantRole | The role played by this colorant.<br>This value is a type 1 enum.<br>1 Other<br>2 (reserved)<br>3 Process<br>4 Spot |
| Value | Read-Write | String(63) | | The name of the color of this colorant using standardized string names from ISO 10175 (DPA) and ISO 10180 (SPDL) which are:<br>**other**<br>**unknown**<br>**white**<br>**red**<br>**green**<br>**blue**<br>**cyan**<br>**magenta**<br>**yellow**<br>**black**<br>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 sub-unit 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:

# 9. 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 |
| DefaultIndex | Read-Write | Integer32 | | The value of **prtMediaPathIndex** corresponding to the default media path; that is, the selection of the default media path. Must be 1 or greater. |
| MaxSpeedPrintUnit | Read-Only | Enum (1) | SpeedPrintUnit | The unit of measure used in specifying the speed of all media paths in the printer.<br>This value is a type 1 enum.<br>1 (reserved)<br>2 (reserved)<br>3 Ten Thousandths Of inches per Hour<br>4 Micrometers per Hour<br>5 Characters per Hour<br>6 Lines per Hour<br>7 Impressions per Hour<br>8 Sheets per Hour<br>9 Dot Row per Hour<br>16 Feet per Hour<br>17 Meters per Hour |
| MediaSizeUnit | 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.<br>This value is a type 1 enum. |

Notes:

## 9.1. Media Path Table - Mandatory

| **prtMediaPath** | **Mandatory** | | **prtMediaPathTable** | |
|---|---|---|---|---|
| Object Name | Access | Type | Value | Description |
| Index | 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. Must be 1 or greater. |
| MaxSpeed | Read-Only | Integer32 | | The maximum printing speed of this media path expressed in prtMediaPathMaxSpeedPrintUnit 's. A value of (-1) implies 'other',. |
| 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 'unlimited'. A value of (-2) implies 'unknown'. |
| 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 (-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 (-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.<br>This value is a type 2 enum.<br>1 Other<br>2 Unknown<br>3 Long Edge Binding Duplex<br>4 Short Edge Binding Duplex<br>5 Simplex |
| Description | Read-Only | String(255) | | The manufacturer-provided description of this media path in the localization specified by **prtGeneralCurrentLocalization**. |

| | | | | |
|---|---|---|---|---|
| Status | Read-Only | Enum (1) | SubUnitStatus | The current status of this media path.<br>This value is a type 1 enum. |

Notes:

# 10. Channel Group - Mandatory

Channels are independent sources of print data. Here, print data is the term used for the information that is used to construct printed may have both data and control aspects. The output of a channel is in a form suitable for input to one of the interpreters as a stream. may be independently enabled (allowing print data to flow) or disabled (stopping the flow of print data). A printer may have o channels.

Basically, the channel abstraction is intended to cover all the aspects of getting the print data to an interpreter. This might include t the data from one place to another, it might include (invisible) compression, it might include encoding or packetizing to provi information sources over a single physical interface and it might include filtering characters that were destined for another kind of c of these aspects are hidden in the channel abstraction.(Note some Page Description Languages have compression built into them so compression refers to compression done by the transport medium and removed before the data is presented to the interpreter.)

There are many kinds of channels;some of which are based on networks and others which are not. For example, a channel can be parallel) connection; it can be a service, such as the Unix Line Printer Daemon (LPD), offering itself over a network connection (inte could be a disk drive into which a floppy disks with the print data is inserted. Each channel is typically identified by the electronic service protocol used to deliver print data to the printer.

| 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 very meager) protocols are run (e.g or old parallel ports) and there are services which often have elaborate protocols that run over a number of protocol stacks. In the e important is the delivery of print data thru the channel.

The channel sub-units are represented by the Channel Group in the Model. It has a current Control Language which can be used which interpreter is to be used for the print data and to query and change environment variables used by the interpreters (and M Applications). There is also a default interpreter that is to be used if an interpreter is not explicitly specified using the Control Channel sub-units are based on an underlying interface.

## 10.1. The channel table and its underlying structure

The first seven items in the Channel Table define the "channel" itself. A channel typically depends on other protocols and interfaces the data that flows thru the channel. It is necessary to provide control of the (perhaps complex) process by which print data ar interpreter. Control is largely limited to enabling or disabling the whole channel. It is likely, however, that more control of the accessing print data will be needed over time. Thus, the ChannelType will allow type specific data to be associated with each cha ChannelType specific groups in a fashion analogous to the media specific MIBs that are associated with the IANAIfType in the Table). As a first step in this direction, each channel will identify the underlying Interface on which it is based. This is the eighth obj row of the table.

Some examples of the kind of control are where compression or encoding is used; and whether the data is filtered to remove f anomolies such as those created by using MS-DOS/PC-DOS LPT1:.

## 10.2. 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 |
|---|---|---|---|---|
| Index | 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. |
| Type | Read-Only | Enum (2) | ChannelType | The type of this print data channel. This object provides the linkage to ChannelType specific groups that (conceptually) extended tha prtChannelTable with additional details about that channel. Must be 1 or greater.<br>This value is a type 2 enum; see below. |
| ProtocolVersion | Read-Only | String(63) | | The version of the lprotocol used on this channel . The format used for version numbering depends on **prtChannelType.** |
| CurrentJobCntlLangIndex | Read-Write | Integer32 | | The value of **prtInterpreterIndex** corresponding to the Control Langauge Interpreter for this channel. 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). Must be 1 or greater. |
| DefaultPageDescLangIndex | Read-Write | Integer32 | | The value of **prtInterpreterIndex** corresponding to the Page Description Langauge Interpreter for this channel. 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). Must be 1 or greater. |
| State | Read-Write | Enum (2) | ChannelState | The state of this print data channel. The value determines whether control information and print data is allowed through this channel or not.<br>This value is a type 2 enum.<br>1 Other<br>2 (reserved)<br>3 Print Data Accepted<br>4 No Data Accepted |
| 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. A value of zero indicates that no interface is assoicated with this channel. Must be 0or greater. |
| Status | Read-Only | Enum (1) | SubUnitStatus | The current status of the channel.<br>This value is a type 1 enum. |

### 10.3. ChannelType Enumeration (enum type 2)

| Value | Enum symbolic name | Description - becomes a MIB/MIF comment |
|---|---|---|
| 1 | Other | |
| 2 | (reserved) | |
| 3 | ch Serial Port | |
| 4 | ch Parallel Port | |
| 5 | ch IEEE1284 Port | |
| 6 | ch SCSI Port | |
| 7 | ch AppletalkPAP | Appletalk Printer Access Protocol |
| 8 | ch LPD Server | |
| 9 | ch NetwareRPrinter | Novell |
| 10 | ch NetwarePserver | Novell |
| 11 | ch port 9100 | |
| 12 | ch AppSocket | a bi-directional, LPD-like protocol using 9101 for control and 9100 for data; Adobe Systems, Inc. |
| 13 | ch FTP | InterNet File Transfer Protocol, RFC ????, "PUT" to printer |
| 14 | ch TFTP | to printer |
| 15 | ch DLC LLC Port | |
| 16 | ch IBM 3270 | IBM |
| 17 | ch IBM 5250 | IBM |
| 18 | ch Fax | |
| 19 | ch IEEE1394 | |
| 20 | ch Transport 1 | port 35 |
| 21 | ch CPAP | port 170 |
| 22 | ch DCERemoteProcCall | OSF |
| 23 | ch ONCRemoteProcCall | Sun MicroSystems |
| 24 | ch OLE | Object Linking and Embedding?, Microsoft |

| | | |
|---|---|---|
| 25 | ch Named Pipe | ? |
| 26 | ch PCPRINT | Banyan |
| 27 | ch Server Message Block | File/Print sharing protocol used by various network operating systems from BIM, 3COM, Microsoft and others. |
| 28 | ch DPMF | Distributed Print Management Framework, IBM |
| 29 | ch DLLProtocolCall | Microsoft |
| 30 | ch VxDProtocolCall | Microsoft |
| 31 | ch SystemObjectManager | IBM |
| 32 | ch DEC LAT | Digital Equipment Corp. |
| 33 | ch NPAP | Network Printer Alliance Protocol |

Notes:

# 11. 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.

## 11.1. 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 |
| Index | 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.<br>This value is a type 2 enum; see below. |
| LangLevel | Read-Only | String(63) | | 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(63) | | 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(63) | | 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. |
| DefaultOrientation | Read-Write | Enum (1) | MediaOrientation | The current default orientation for this interpreter This value maybeoverridden for a particular job (e.g., by a command in the input data stream).<br>This value is a type 1 enum.<br>1 Other<br>2 (reserved)<br>3 Portrait<br>4 Landscape |
| AddressabilityFeedDir | Read-Only | Integer32 | | The maximum interpreter addressability in the feed direction in 10,000 **prtMarkerAddressabilityUnit** s (see **prtMarkerAddressabilityFeedDir** ) for this interpreter. The value (-1) means other and specifically indicates that the sub-unit places no restrictions on this parameter. |
| AddressabilityXFeedDir | Read-Only | Integer32 | | The maximum interpreter addressability in the cross feed direction in 10,000 **prtMarkerAddressabilityUnit** s (see **prtMarkerAddressabilityXFeedDir** ) for this interpreter. The value (-1) means other and specifically indicates that the sub-unit places no restrictions on this parameter. |
| DefaultCharSetIn | Read-Write | Integer32 | | An index into the CharacterSetTable that identifies the default character set for octets encounterd ourside a context in which the PDL establishs the interpretation of octets |
| DefaultCharSetOut | Read-Write | Integer32 | | An index into the CharacterSetTable that identifies the default character set fordata coming from this interpreter through the printer's output channel |

| | | | | |
|---|---|---|---|---|
| TwoWay | Read-Only | Enum (1) | YesNo | Indicates whether or not this interpreter returns information back across the data channel to the host. Must be 2 or 3.<br>This value is a type 1 enum.<br>1 Other<br>2 (reserved)<br>3 Yes<br>4 No |

## 11.2. IntpLangFam - Interpreter Language Family (enum type 2)

| Value | Enum symbolic name | Description - becomes a MIB/MIF comment |
|---|---|---|
| 1 | other | |
| 2 | (reserved) | |
| 3 | lang PCL | Printer Control Language, or Printer Command Language<br>Starting with version 5, HP-GL/2 is part of PCL.<br>Hewlett-Packard Co. |
| 4 | lang HPGL | Hewlett-Packard Graphics Language. Originally used to control pen movements on plotters. Includes many device specific commands.<br>Hewlett-Packard Co. |
| 5 | lang PJL | Peripheral Job Language. Appears in the data stream between data intended for a page description language.<br>Hewlett-Packard Co. |
| 6 | lang PS | PostScript (TM) Language . PostScript is a trademark of<br>Adobe Systems Incorporated which may be registered in some jurisdictions. |
| 42 | lang PSprinter | The PostScript Language used for control ( with any PDLS)<br>Adobe Systems, Incorporated |
| 7 | lang IPDS | Intelligent Printer Data Stream<br>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.<br>Facilitates system level device<br>control, document tracking and error<br>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.<br>Lexmark International, Inc. |
| 9 | lang EscapeP | |
| 10 | lang Epson | |
| 11 | lang DDIF | Digital Document Interchange Format<br>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.)<br>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)<br>ISO standard, Geneva, Switzerland |
| 15 | lang MODCA | Mixed Object Document Content Architecture<br>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.<br>Pennant Systems, IBM |
| 16 | lang ReGIS | Remote Graphics Instruction Set,<br>Digital Equipment Corp., Maynard MA |
| 17 | lang SCS | SNA Character String<br>Bi-directional print data stream for SNA LU-1 mode of communications<br>IBM |
| 18 | lang SPDL | ISO 10180 Standard Page Description Language<br>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.<br>QMS, Inc., Mobile AL |

| | | |
|---|---|---|
| 23 | lang DSC DSE | DSC-DSE: Data Stream Compatible and Emulation<br>Bi-directional print data stream for non-SNA (DSC) and SNA LU-3 3270 controller (DSE) communications<br>IBM |
| 24 | lang WPS | Windows Printing System, Resource based command/data stream used by Microsoft At Work Peripherals.<br>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.<br>QMS, Inc. Mobile, AL |
| 28 | lang CPAP | Common Printer Access Protocol<br>Digital Equipment Corp. Maynard MA |
| 29 | lang DEC PPL | Digital ANSI-Compliant Printing Protocol (DEC-PPL)<br>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)<br>ISO standard, Geneva, Switzerland |
| 31 | lang NPAP | Network Printer Alliance Protocol<br>IEEE 1284.1 |
| 32 | lang DOC | Document Option Commands, Appears in the data stream between data intended for a page description language.<br>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).<br>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<br>Asian countries than in US.<br>NEC |
| 35 | lang NPDL | Page printer language for Japanese market.<br>NEC |
| 36 | lang 201PL | Serial printer language used in the Japanese market.<br>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<br>IBM Japan |
| 39 | lang LIPS | Canon |
| 40 | TIFF | A format for image data<br>Aldus |
| 41 | Diagnostic | A Hex dump of the input to the interpreter |

Notes:

*1. references to resolution are removed from max addressability objects; the names of these objects are changed to match the similar objects in the Marker Table; unknown is not allowed for max addressability (since it is a read-only value); and -1 was used to indicate no restriction on the resolution value (both 0 and -1 seemed to be doing the same thing in the previous description.)*

## 12. 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. This localization applies both to the actual display on the console as wellas the encoding of these console objects/attributes in management applications. |
| NumberOfDisplayLines | Read-Only | Integer32 | | The number of lines on printers physical display. This value is 0 if there are no lines on the physical display or if there is no physical display. Otherwise, it must be 1 or greater. |
| NumberOfDisplayChars | Read-Only | Integer32 | | The number of characters per line displayed on the physical display. Must be 1 or greater. |
| Disable | Read-Write | Enum (2) | DisableState | This object enables or disables manual input from the operators console.<br>This value is a type 2 enum.<br>1 Other<br>2 (reserved)<br>3 Enabled<br>4 Disableed |

Notes:

### 12.1. Display Buffer Table - Mandatory

| **prtConsole Mandatory prtDisplayBufferTable** | | | | |
|---|---|---|---|---|
| Object Name | Access | Type | Value | Description |
| DisplayBufferIndex | Read-Only | Integer32 | | A unique value for each console line in the printer. The value is used to identify this console line. Although these values may change due to a major reconfiguration of the device (e.g. the addition of new console linesto the printer), values are expected to remain stable across successive printer power cycles. |
| 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. Writing a zero length string unlocks that line. Whether the Printer can overwrite a line before it is unlocked is up to the vendor. Printer generated strings shall be in the localization specified by prtConsoleLocalization. Management Application generated strings should be localized by the Management Application. |

Notes:

| **prtConsole Mandatory prtConsoleLightsTable** | | | | |
|---|---|---|---|---|
| Object Name | Access | Type | Value | Description |
| LightsIndex | 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. If both prt ConsoleOnTime and prtConsoleOffTime are 0 then the light is always off. |

| | | | | |
|---|---|---|---|---|
| Color | Read-Only | Enum (2) | LightColor | The color of this light.<br>This value is a type 2 enum.<br>1 Other<br>2 Unknown<br>3 White<br>4 Red<br>5 Green<br>6 Blue<br>7 Cyan<br>8 Magenta<br>9 Yellow |
| Description | Read-Only | String(255) | | The vendor description or label of this light in the localization specified by ConsoleLocalization. |

Notes:

## 13. 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.

### 13.1. Alerts Table - Mandatory

| **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 is an incrementing integer starting from zero every time the printer is reset 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 a reset of the printer shall have an index value of 1.<br>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 known index value with the current highest index value. The management application will then update its copy of the alert table.<br>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. The management station can detect when binary alerts have been deleted by requesting an attribute of each alert, and noting alerts as deleted when that retrieval is not possible. |
| 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.<br>1 Other<br>2 (reserved)<br>3 Critical<br>4 Warning |

| | | | | |
|---|---|---|---|---|
| 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:<br>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.<br>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.<br>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.<br>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.<br>This value is a type 2 enum.<br>1 Other<br>2 Unknown<br>3 Untrained<br>4 Trained<br>5 Field Service<br>6 Management |
| 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.<br>This value is a type 1 enum.<br>1 Other<br>2 (reserved)<br>3 Input<br>4 Output<br>5 Finisher<br>6 Media Path<br>7 General Printer<br>8 Marker<br>9 System Controller<br>10 Interface<br>11 Interpreter<br>12 Console<br>13 Channel |
| GroupIndex | Read-Only | Integer32 | | An index of the row within the principle table in the group identified by **prtAlertGroup** that represents the sub-unit of the printer that caused this alert. The combination of the Group and the GroupIndex defines exactly which printer sub-unit caused the alert.; for example, Input #3, Output #2, and Marker #1. |
| Location | Read-Only | Integer32 | | The sub-unit location that is defined by the printer manufacturer to further refine the location of this alert within the designated sub-unit. The location is used in conjunction with the Group and GroupIndex values; for example, there is an alert in Input #2 at location number 7. |
| AlertCode | Read-Only | Enum (2) | *(sub-unit)* AlertCode | The code that describes the type of alert for this entry in the table. There are different codes for each sub-unit type; for example, Media Supply Low and Media Supply Empty are Alert codes for the Input sub-unit<br>This value is a type 2 enum; see below. |

| | | | | |
|---|---|---|---|---|
| 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. |

## 13.2. Alert Codes

| | | |
|---|---|---|
| InputAlertCode (enum type 2)<br>1 Other<br>2 Unknown<br>3 Cover Open<br>4 Cover Closed<br>5 Interlock Open<br>6 Interlock Closed<br>7 Configuration Change<br>8 Jam<br>9 Media Tray Missing<br>10 Media Size Change<br>11 Media Weight Change<br>12 Media Type Change<br>13 Media Color Change<br>14 Media Form Parts Change<br>15 Media Supply Low<br>16 Media Supply Empty | OutputAlertCode (enum type 2)<br>1 Other<br>2 Unknown<br>3 Cover Open<br>4 Cover Closed<br>5 Interlock Open<br>6 Interlock Closed<br>7 Configuration Change<br>8 Jam<br>9 Media Tray Missing<br>10 Media Tray Almost Full<br>11 Media Tray Full | MarkerAlertCode (enum type 2)<br>1 Other<br>2 Unknown<br>3 Cover Open<br>4 Cover Closed<br>5 Interlock Open<br>6 Interlock Closed<br>7 Configuration Change<br>8 Jam<br>9 Toner Empty<br>10 Ink Empty<br>11 Print Ribbon Empty<br>12 Toner Almost Empty<br>13 Ink Almost Empty<br>14 Print Ribbon Almost Empty<br>15 Waste Toner Receptacle Almost Full<br>16 Waste Ink Receptacle Almost Full<br>17 Waste Toner Receptacle Full<br>18 Waste Ink Receptacle Full<br>19 Fuser Under Temperature<br>20 Fuser Over Temperature<br>21 OPC Life Almost Over<br>22 OPC Life Over<br>23 Developer Almost Empty<br>24 Developer Empty |
| GeneralPrinterAlertCode (enum type 2)<br>1 Other<br>2 Unknown<br>3 Cover Open<br>4 Cover Closed<br>5 Interlock Open<br>6 Interlock Closed<br>7 Door Open<br>8 Door Closed<br>9 Power Up<br>10 Power Down | MediaPathDeviceAlertCode (enum type 2)<br>1 Other<br>2 Unknown<br>3 Cover Open<br>4 Cover Closed<br>5 Interlock Open<br>6 Interlock Closed<br>7 Configuration Change<br>8 Jam<br>9 Media Tray Missing<br>10 Media Tray Almost Full<br>11 Media Tray Full | |
| InterpreterAlertCode (enum type 2)<br>1 Other<br>2 Unknown<br>3 Configuration Change<br>4 Memory Increase<br>5 Memory Decrease<br>6 Cartridge Added<br>7 Cartridge Deleted<br>8 Resource Added<br>9 Resource Deleted<br>10 Resource Unavailable | InterfaceAlertCode (enum type 2)<br>1 Other<br>2 Unknown<br>3 Configuration Change | ChannelAlertCode (enum type 2)<br>1 Other<br>2 Unknown<br>3 Configuration Change |

Notes:

## Appendix A - Glossary of terms

Addressability -- on the marker, the number of distinctly setable marking units (pels) per unit of addressability unit; for example, 300 dots per inch is expressed as 300 per 10,000 Ten Thousandths Of Inches and 4 dots per millimeter is 4 per 10,000 Micrometers. Addressability is not resolution because marks that are one addressability position apart may not be independently resolvable by the eye due to factors such as gain in the area of marks so they overlap or nearly touch.

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

Bursting -- the process by which continuous media is separated into individual sheets, typically by bursting along pre-formed perforations.

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.

Collation -- in multiple copy output, placing the pages from separate copies into separte output bins

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

Decollating -- the process by which the individual parts within a multi-part form are separated and sorted into separate stacks for each part.

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

## Appendix B - Media Size Names

For the convenience of management application developers, this appendix lists the standardized media size names from ISO/IEC 10175 Document Printing Application (DPA). Management applications that present a dialogue for choosing or displaying media size are encouraged to present relevant names from this list to avoid requiring the user to remember the physical dimensions used to describe the size of the media. A printer implementing the Printer MIB/MIF has no knowledge of these names, however; all media sizes in the MIB/MIF are given in terms of media dimensions as the values of **prtInputChosenMediaDimFeedDir** and **prtInputChosenMediaDimXFeedDir**.

| 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 |

## Appendix C -Media Names

For the convenience of management application developers, this appendix lists the standardized media names from ISO/IEC 10175 Document Printing Application (DPA). Management applications that present a dialogue for choosing media may wish to use these names as an alternative to separately specifying, size, color, and/or type. 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. If selection of media by name is used, the attributes (size, type or color) implied by the name must be explicitly mapped to the appropriate attributes/object (**prtInputDeclaredMediaDimFeedDir**, **prtInputDeclaredMediaDimXFeedDir, prtInputMediaType** and **prtInputMediaColor**) in the MIF/MIB. The attribute/object **prtInputMediaName** is intended for display to an operator and is purely descriptive. The value in **prtInputMediaName** is not interpreted by the printer so using a standard name for this value will not change any of the other media attributes nor will it cause an alert if the media in the input sub-unit does not match the name.

| 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 |

# Appendix D - Roles of Users

## 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.

## 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.

### 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.

### 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 |

## Appendix E - Draft Specification for the Finishing Sub-unit

The following text is an initial proposed draft for the management of a Finisher sub-unit. Further work on this, including complete replacement of the proposal, has been delegated to the Finisher Working Group.

## 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.

The relationship between a finisher and the output sub-unit depends on the type of finishing process and the exact implementation of the printer system. The Finishing standard allows for the logical association of a finishing process with an output sub-unit but does not put any restrictions on the exact sequence or interaction with the associated output sub-unit. 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 sub-unit may be associated with multiple finishing sub-units and a single finishing sub-unit may be associated with multiple output sub-units.

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

### Output Associated Finishers Table - Optional

| **prtOutput - Associated Finisher Optional** | | | **prtOutputAssocFinisher** | |
|---|---|---|---|---|
| Object Name | Access | Type | Value | Description |
| Index | Read-Only | Integer32 | | The Index of the output sub-unit to which the finisher is connected. |
| Stitching | Read-Only | Enum (1) | YesNoOnOff | By its presence, this object indicates that the output sub-unit 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 sub-unit; the value serves as the index for identifying the sub-unit 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 sub-unit 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 sub-unit; the value serves as the index for identifying the sub-unit 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 sub-unit 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 sub-unit; the value serves as the index for identifying the sub-unit 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 sub-unit 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 sub-unit; the value serves as the index for identifying the sub-unit 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 sub-unit 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 sub-unit; the value serves as the index for identifying the sub-unit 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 sub-unit 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 sub-unit; the value serves as the index for identifying the sub-unit in the **prtFinisherTable**. This value is a type 1 enum. |

Notes:

### Finishing Configuration - Optional

The finishing table maintains mappings between configurations of possibly zero or more finishers and zero or more output sub-units. Each output sub-unit 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 |
| DefaultIndex | Read-Write | Integer32 | | The Index of the default finisher sub-unit, i.e., the value of the corresponding finisher sub-unit's **Index** value. A value of 0 means that a finisher is not used in default configurations. |

Notes:

### Finisher Table - Optional

| **prtFinisher** | **Optional** | | **prtFinisherTable** | |
|---|---|---|---|---|
| Object Name | Access | Type | Value | Description |
| Index | Read-Only | Integer32 | | A unique value used by the printer to identify this finisher sub-unit. Although these values may change due to a major reconfiguration of the device (e.g. the addition of new finisher sub-units 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 sub-unit. This value is a type 2 enum. |
| NamedFinishing | Read-Only | Enum (2) | NamedFinishing | The finishing operations available through this finishing sub-unit. This value is a type 2 enum. |
| ConsumableUnit | Read-Only | Enum (1) | FinisherConsumableUnit | The consumable resource unit appropriate to this finisher sub-unit 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) | SubUnitStatus | The current status of this finisher. This value is a type 1 enum. |

Notes:

### Finisher Enumerations

| 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 |

| 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 |

| FinisherConsumableUnit (enum type 1) | |
|---|---|
| 1 | Other |
| 2 | Unknown |
| 3 | Thousandths Of Inches |
| 4 | Micrometers |
| 8 | Sheets |
| 12 | Thousandths Of Ounces |
| 13 | Tenths Of grams |
| 14 | Hundredths Of fluid ounces |
| 15 | Tenths Of milliliters |
| 16 | Feet |
| 17 | Meters |
| 18 | Each |

| 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 |

## Appendix F - 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

## 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?

   CLOSED: 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"?

   CLOSED: 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?

   CLOSED: 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?

   CLOSED --- Current set of objects are deemed necessary

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?

   OPEN: It was agreed that "community names" was not a good solution to the problem of multiple printers accessed via one agent. If there was a way to integrate the use of an additional index on all the printer objects with the existing definitions of MIB-II and the Host Resources MIB, then the additional index approach might be favored. If no such method is found, however, the use of contextLocalEntity in SNMPv2 seems to be the favored approach. How would this translate into DMTF?

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?

    CLOSED: SNMP doesn't normally use "SNAPing". Some of the integer objects' maximums and minimums are available as other objects. Those that are not available can be limited via RANGE values within the MIB that is shipped with the printer. All invalid values (both ENUMS and INTEGERS) will be rejected. The issue will be re-opened pending a review of the number of R/W integers that do not have mins/maxs available as another object.

    There is one case where SNAPing is desired. That is in Media Dimensions. For this case (and similar cases, but no others identified), the SNMP approach is to have 2 objects per value. The first object is the requested or declared value; the second object is the granted (SNAPed) value. For media size the declared value may either be sensed by the printer or written from a (remote) Control Panel or Mgt. Application. The granted value is the value the printer assumes in its imaging operations. The granted value must be set immediately in response to setting the declared value.

    See also the resolution of issue 65.

13) Are "Input - dimensions extended" mandatory?

    CLOSED: The table extension has been removed; the printable area information in this table was moved to the Marker class and changed margin requirements that apply to all paper types.

14) In what units is printable area reported?

    CLOSED: .0001 inches or .001 mm

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

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

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

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

CLOSED: Outputs: Association of outputs to finishers is delegated to the Finisher Working Group.

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

CLOSED: Yes, but because imageable area also depends on media size and device resolution, it was decided that only margin information would be reported only on a marker basis and would specify margin sizes that would guarantee printing on any media size.

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

CLOSED - Yes, we will address this on a case by case basis; 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?

CLOSED - see #68 for closure information. Îf 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?

CLOSED: There is a hierarchy used to determine the interpreter to use:

1) If an interpreter is specified in the data stream, use that interpreter

2) Otherwise use the default for the channel

3) Otherwise use the overall default interpreter

4) Otherwise the default interpreter is undefined or manufacturer specific

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

CLOSED: 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?

CLOSED: They refer to the physical size of the console.

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

CLOSED: 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?"

CLOSED: Yes, we need an index column in each table. We should name all the fields now called "ID" "Index". The document has been changed to reflect this resolution. This includes standardizing all the references to Indexes to the form "The value of XXXXIndex corresponding to ..."

26) Are all consoles using the same language?

CLOSED: 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?

CLOSED: 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?"

CLOSED - see 67) for closure information.

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) How should channel relate to underlying layers and to other existing MIBS?

CLOSED: Additional details for channels will be specified in separate groups one per channel type and the association to these groups will be based on the association of the channel type enumeration in the same way that the association of additional details (media specific MIBs) on interfaces is based on the interface type (IANAIfType) in the interface table (reference by analogy)- There is also a pointer to the interfaces MIB to identify the underlying interface. While this is not a complete solution, it is sufficient for this implementation. (Some of the other issues raised but not completely answered were: should it be a more complex implementation; should we add a "Channel Name" object?)

"A channel is an independent source of print data which can be enabled or disabled, is directable to an interpreter, and its output is suitable for the interpreter."

(There was a long discussion on how channels are linked to the underlying layers. This discussion included the issue of how to deal with a WPS printer: where is the channel - within the PC or across the 1284 parallel port to the physical printer? In order to restrict the scope of the work, the group will focus on the physical printer device. Environments like WPS and the SUN printer will fit in the model but the details of that management implementation is up to the manufacturer.)

30) How well does the Host MIB work for the system controller?

CLOSED - device and storage groups of the host resources MIB will be adequate. The HOST MIB is probably overkill for a printer but the work needed to get the SNMP community to agree to have new objects in the printer MIB that largely duplicate host MIB objects is too great.

(Is there work in progress to create a MIF which corresponds with the HOST MIB? Since the channel group uses the interfaces table from MIB II, how is this done with DMTF?)

31) How much of the Host MIB is mandatory for the system controller?

CLOSED - The hrStorage and hrDevice groups, as defined in the Host MIB (RFC1514), are required for this MIB/MIF; see also 30).

32) What is the difference between INTEGER and INTEGER32?

CLOSED: INTEGER is used for ENUMs and INTEGER32 for values.

33) Do we need and "everything back to OK status" alert?

CLOSED: No, applications should scan the alert table.

34) Should the console be writable?

CLOSED: Yes: Addressed by issues #22 and #23 and therefore closed.

35) Should Company names be included in channels? Interpreters?, etc?

CLOSED Yes - to exclude them would make some of the names less clear.

Discussion occurred on whether to make the channel less ambiguous by adding revision, version, etc. object. It was agreed to add a string to allow each manufacturer to uniquely identify specific version/lexvel of a channel.

36) Should objects in the channel group relate to the ISO seven layer model?

CLOSED: NO, No attempt will be made to correlate the channels to the ISO model.

37) Should the default orientation be by interpreter or overall?

CLOSED: Overall only.

38) Should we include an operator panel lock object?

CLOSED: Yes

39) Is the description field of the alert table localized when inserted in the table or when the table is accessed?

CLOSED: This is vendor implementation specific. The recommendation of the group was that the best implementation would be to localize when the table is accessed.

40) 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?

    CLOSED: There will be only one naming scheme and it will be used for both DMTF and SNMP and will follow the syntax restrictions of both DMTF and SNMP syntaxes.

41) 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**?

    CLOSED, see 40)

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

CLOSED: will have two sizes of strings: 255 for long strings including descriptor strings; 63 for shorter strings including standardized string values and for Version and LangLevel. This is a change from prtspec1.doc.

43) 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.

44) Should we change the name from “Skins” to “Doors” to be more PC?

CLOSED: we will use "Cover"

45) Should MediaType and MediaColor string value names be localized?

CLOSED: NO, they should be expressed in NVT ASCII

46) 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?

CLOSED: see 45)

47) Is the prefix of **lang** and **ch** 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?

CLOSED: YES, we will use the abbreviations agreed in Washington.

48) The LineStatus enum isn’t used by any object/attribute. Should ChannelStatus use LineStatus, instead of SubUnitStatus?

CLOSED: NO, and line status will be eliminated

49) 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?

CLOSED: The Storage Group and Device Group of the Host MIB are useful to printers (and not unduly burdensome). The System Group, on the otherhand, is difficult for printers to implement because many of the values are irrelevant or in the case of times printers often do not have clocks. Hence, support of the Systems Groups is not required. We need implement only the Storage and Device Groups of the MIB. Steve W. will explore making changes to the Host Resources MIB to reflect a reduced lcompliance requirement. Each manufacturer will have to make a determination of what to include.

From the MIB 2, only the SYSTEM and INTERFACES group need to be supported to be compliant with this MIB.

50) 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?

   CHANGE, MrkSupplyType become type 2.

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

   NO CHANGE, left as type 3

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?

   NO CHANGE, additional colorant roles occur infrequently and the Marker Table has an object for each of the existing roles so adding a role potentially changes the Marker Table, this enum is left as a type 1

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?

   CHANGE, ChannelState becomes type 2 to allow someone to implement separation of command and data.

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?

   NO CHANGE, because it maps so closely to the model..

51) 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?

CLOSED, see 40) above.

52) Community Names, are they a solution the to problem of having one agent for several printers

OPEN - Community names wouldn't work with Host based printing where the community name applies to the whole PC not just the printer. Steve W. will attempt to solve the problem with printer MIB and the HOST MIB using an index otherwise we will go with an SNMP v2 only solution for the multiple printers on a single IP address. See also 8).

53) Should the finishers group be removed and left to a finishers Working Group?

CLOSED: YES, the Finisher Group is removed to an appendix and left to a working group of Finisher vendors to define both the Finishing objects and their relationship to the remaining Printer Model.

54) Should strings be DisplayStrings, OCTETs or something else?

CLOSED: For localized strings, use the InternationalDisplayString type of the HostMIB, but reference our localization tables and objects. For non-localized strings (such as MediaColor) use DisplayString.

55) Should the console object be dropped from the MIB

CLOSED: Keep the console. Do not include key press function.

56) Should a "current operator" (a person) object be added to the PrinterGeneral class?

CLOSED: Use the SysContact object in the System Group of MIB-II

57) Add TIFF to interpreters?

CLOSED: Add a TIFF Interpreter to the Interpreter table, Zilles to provide information on TIFF

58) Add Diagnostic/Hex Dump interpreter?

CLOSED: Joel will add to the interpreter enum list.

59) Should we put out a Press Release when the Printer MIB initial draft, final draft are posted to IETF?

CLOSED: Discussion on need for a press release in conjunction with the release of the MIB and MIF. Discussion on schedule and the impact of hitting or missing the schedule.

60) All references to the ISO OSI seven layer model should be removed

CLOSED: All such references have been replaced by other text, but the group took no position on the advisability of using the seven layer model.

61) Update Channel Type table. Replace OS/2 LAN SERVER with System Message Block and add DLL, VxD, DCE RPC, ONC RPC.

CLOSED: Done.

62) What DMI class name should we use: DMTF?, DMTF Printer Working Group?, Printer Working Group?, Joint Printer Working Group?

CLOSED: DMTF/IETF Printer Working Group.

63) Is the Printer Management Model adequate to model host based printing solutions?

CLOSED - We believe that our printer model covers the case of host based printers. We will leave in the various Microsoft requested channels applicable to host based printing.

64) Is the Printer Management Model adequate to model

CLOSED - We believe our printer model covers (with extensions) scanner/copier/fax multi-function devices. Scanner may have an additional media path, fax might have additional channels, etc.

65) How is the range of legal values determined for Integer32 variables

CLOSED - We reviewed the R/W Integer objects that do not inherently have a range. The only objects which need a range were in the input group: *Declared*MediaDimXFeedDir and *Declared*MediaDimFeedDir. We discussed and agreed to use the SNMP method for doing something like a snap but using two objects. The new objects are read-only and called *Granted*MediaDimXFeedDir and *Granted*MediaDimFeedDir

The remaining Interger32 objects were reviewed to make sure the description provided reasonable range indications.

66) Do we need to add a version string to the channel group?

CLOSED: We will add a string to allow a manufacturer to further identify the channel.

67) Are other and unknown valid for integers?

CLOSED: We will review and include on a case by case basis.

68) Should we add an "unitialized" state to go along with "other" and "unknown"?

CLOSED: No one in attendance made a case to add the "unitialized" state. (see, however, issue 93 below)

69) Remove prt Output Slipsheet from prt Output Table Augment and refer it to the Finisher group

CLOSED: Agreed, since it needs supplies, it is suggested that it be grouped with similar things, such as Stapling.

70) It is difficult to associate enumerations with where they are used.

CLOSED: enumerations used only once will be put with place of use. Common enumerations will be in one place

71) Better documentation of values Critical and Non-Critical in the Sub Unit Status enumeration is needed

CLOSED: comments have been added to values 4,5, and 6.

72) Since Hours is now the only time unit allowed for SpeedTimeUnit in the MediaPath Group, why not fold the time unit into the enumeration for SpeedPrintUnit.

CLOSED: The only SpeedTimeUnit value is added to the SpeedPrintUnits, e.g. Ten Thousandths of an Inch per Hour; Feet (16) and Meters(17) are added to the enumeration (due to overflow of with Ten Thousandths of an Inch per Hour) and the MaxSpeedTimeUnit object is removed from the Media Path Group.

73) Media paths: what is the interpretation of the Media Path Type

CLOSED: The idea behind Media Paths was to allow a logical description of the possible flows through through the marker in the form of a table. The existance of a particular logical path, such as long edge duplex ro short edge duplex does not imply that the path is realized as a physical entity, but that there is a path the acheives the named effect. For example, the two forms of duplexing may be realized on the same physical paper flow by rotating the output of the interpreter.

Remove the Either Edge Duplex value from the enumeration; it does not describe a desired effect.

74) Media Orientation: (a) need description of a frame of reference to give meaning to Portrait and Reverse Portrait

CLOSED: remove the Reverse options because they require loading the paper correctly to give them meaning and the meaning may be different for different interpreters.

75) Should Media Orientation be an attribute of an Interpreter rather than the Printer as awhole.

CLOSED: Yes, move Default Orientation from prtGeneral to prtInterpreterTable. This allows interpreter to report its default orientation.

76) Are the rules for what should be read/write correct; for example, must the printer process the value written and behave differently? This does not seem to be thecase with Name in Input Group Type Extended?

OPEN: There is value in being able to name sub-units (or parts thereof), such as input feeders, to allow users to recognize that the device is present and therefore use it. These names are typically assigned by an administrator and are may be written into the MIB/MIF if not recognized by the device.

77) Output bin size is read/write. Are there cases where there would be difference among the set of output bins

CLOSED: no change to the current spec.

78) Some changes are probably needed for the MIB to match the MIF model in which there is an uniform algorithm for mapping SNMP requests to the DMI.

OPEN: we need to put the 14 current scalers into one or more tables; we probably need to add a component id index column on all tables

79) Status

CLOSED (mostly): We accept the concept of DeviceStatus as a device independent Red, Yellow, Green status indicator. We can accept PrinterStatus if some values (to be supplied by Zilles) are added. We have some reservations on how good a solution the PrinterDetectedErrorState is. Some would prefer it be deprecated and mgt appl use the printer MIB alert table instead.

80) what is called System Message Block should be called Server Message Block

CLOSED: done

81) The enumerations for channels and interpreters in the Printer MIB do not have the agreed prefix (ch and lang) which allow trademarks to be spelled correctly (where case matters).

CLOSED: The prefixes "ch" and "lang" were reaffirmed and chNPAP was add it as number 33 in the Channel Type enumeration.

82) We need a more precise explaination of configuration change.

CLOSED: add text simiilar to the following:

Mangement applications may cache iinfrequently changed configuration information about the subunits on the printer. This object should be incremented whenever this information is changed thereby triggering the application to re-acquire the information. For example, this counter would not be increment when the level of an input device changes or when an input tray is removed. This object would be incremented when an input tray that has different media than was previously there is inserted

83) How should trademarks and registered trademarks be handled in the Printer Working Group ducuments?

CLOSED: "All Trademarks and Registered Trademarks are the property of their respective owners." will be added to the internet draft and other documents.

84) Are all the use of -1 (for other, unlimited, ...) valid

CLOSED: the -1 value will be remove from MediaWeight

85) More detail is needed on what is controlled by DefaultCharSetIn

CLOSED: the text is modified to read "An index into the CharacterSetTable that identifies the default character set for octets encounterd ourside a context in which the PDL establishs the interpreation of octets ."

86) Remove "other" from the enumerations for PageDeliveryOrientation and StackingOrder

CLOSED: done

87) Correct the units for addressability from Thousandths of an Inch to Ten Thousandths of an Inch and fix the multipliers in the descriptions of Addressability to 10,000 (from 1000)

CLOSED: done

88) How should the character set ids be standardized

CLOSED: They will be specified with a type 3 enumeration which will be seeded with values provided by Zilles (for standard char sets) and Lewis and Gyllenskog (for vendor char sets). ASCII will not be used as an enumeration value, but may be reported by a management appl.

89) Must the printer always provide a value for LifeCount and PowerOnCount?

CLOSED: Yes, if it wants to claim conformance. LifeCounter and PowerOnCounter cannot be negative. We will not allow -1, -2 values.

90) Should there be separate control of the localization of the console seen on the console and seen by a mgt appl?

CLOSED: No, the current structure remains; it was recognized that this requires the mgt appl to change what a console viewer sees to change the localization to a language the person using the mgt appl wants.

91) Can we add objects for Current Operator and Person Who Last Serviced Printer?

CLOSED: add a new optional group with these two objects will be added

92) Should we add a distinguish value for non-zero capacity/content

CLOSED: Yes, the value -3 will be added as an allowed value for prtInputCurrentLevel, prtOutputRemainingCapacity and prtMarkerSuppliesLevel

93) Should there be an "uninitialized value" in writeable objects

CLOSED: Because an object may be writable even after it has a value other than "uninitialized", it is necessary for a Mgt Appl to test writability to determine whether it should encourage the user to set a value. This is especially true for printers that can detect the presence of paper, but not how much (see 93 above). It was noted that the group very much appreciates Tom Hastings work on this issue even though we did not choose his proposed solution.

94) Revise the description of MediaName to make it clear that the value of this object is for informational purposes only and the printer will not interpret the value in any way

CLOSED: Agreed, Zilles to provide text for the description and intros to appendixes B and C.

95) Remove prt Output Job Collated from prt Output Table Augment

CLOSED: Agreed, it has disappeared from DPA already

96) As currently documnented, prtInterpreterDefaultIndex will never be used because each channel must have a valid default interpreter.

CLOSED: remove the prtInterpreterDefaultIndex from the Interpreter Group and report the default interpreter in the Channels table; if there is only one default interpreter all channels will report the same value.

97) There should be prtInterpreterTwoWay object in the Channels Table

CLOSED: Not accepted; no strong benefit to Mgt Appls was identified and the printers already know whether the channel is uni or bi-directional.

98) The current definition of writing to a line in the console buffer locks that line unitl a zero length string is written, locking out any use of the line by the console code. This seems too restrictive.

CLOSED: The description was changed to say that writing a 0 length string will unlock the line, but that the vendor was responsible for deciding whether and how action at the console itself may affect the buffer is up to the vendor to decide. In particular, console action may grab a buffer line to display a menu.

99) The comment on the AppSocket type in the Channels Table needs clarification

CLOSED: revised text was accepted.

100) The value "other" is not meaningful for MaxMediaXFeedDir and for both MinMediaDirs:

CLOSED: the "other(-1)" is changed to "unknown(2)" for these three and the value associated with -1 for MaxMediaFeedDir is interpreted as being "unlimited" and "unknown(-2)" is allowed.

101) How will SNMP optional table extensions be handled in the MIF which does not provide for extensions of a group (optional or otherwise)?

CLOSED: All SNMP groups that form a single Table will be merged into a single DMTF Group whether optional or mandatory, but it is acceptable to respond "Unknown"

102) Should we endorse some interlock method?

CLOSED: No. A method for timed advisory locks was proposed by Steve Waldbusser. It was decided that the Printer MIB was not the place to introduce a new technology that might be needed by other MIBs as well. In addition, no clear need for the lock was established. (For example, it was thought that clashes on the localization of the operator console would be very infrequent). Therefore, no method for locking to resolve clashses among management applications was endorsed.

103) The description of the console lock is incorrect and does not correspond to the description in PRT_OBJ7.DOC

CLOSED: it was agreed that prtConsoleLock would be changed to prtConsoleDisable and the old text, perhaps editted for clarity, would be used. This text specifies that the object disables the operator panel, not other mgt applications.

104) It should be clear that when pruning the alert table simple alerts should be eliminated before binary alerts

CLOSED: The rules for maintaining the alert table were changed to say that simple change events (e.g. configuration changes) are removed before non-critical binary change events which are removed before critical change events are removed. Within each class the vendor is free to choose which alert entires are to be removed.

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

### Changes from prtspec Version 0.1, 15-May to Version 0.2, 31-May

1. Change the headings for MarkerSupplies and MediaPath Tables from Optional (a mistake) to Mandatory, matching the table itself.
2. Change MaxCapacity and Level in the MarkerSupplies Table from read-only to read-write to allow an operator to declare what is loaded. Change Value in the MarkerColorant Table from read-only to read-write to allow spot colors to be declared.
3. Split MediaDim(X)FeedDir into two pairs of objects: DeclaredMediaDim(X)FeedDir and GrantedMediaDim(X)FeedDir.
4. Add double line bottom separator to indicate where the index objects (above the double line) are for each table.
5. Change ID to Index as per the SNMP convention and add Indexs for Localization, CharacterSets, DisplayBuffer.
6. Update Channel Group description from latest Channel Group proposal. Add Protocol Version to allow further refinement of the ChannelType where relevant.
7. Move Finisher Group to an Appendix and defer all futher development of this group and its association to the Output Group to the Finisher Working Group. Finishers were left in the model, however.
8. Revised picture of the model to show only entities that correspond to Groups (e.g., not Localization or CharacterSets which in in the General Printer Group). Reordered the elements so they will correspond to the order of presentation agreed in Pittsburgh.
9. Attempted to ensure consistent usage of "sub-unit", any part of the printer that has an associated table, and "device", the whole printer.
10. Made Output units objects Read-Write to correspond to the similar Input units objects..
11. Began implementation of the single name for both SNMP and DMTF policy (see issue 40).
12. Provided text for the System Group of MIB-II and the System Controller based on the resolution of issue 49).
13. Standardized references to other tables in the form "The value of **prtXIndex** corresponding to ...", where **X** is the name of the relevant index in the table being referenced. This applies to CurrentLocalization, Localization CharacterSet, InputDefaultIndex, ChannelCurrentJobCntlLangIndex, ChannelDefaultPageDescLangIndex., InterpreterDefaultIndex, ...
14. Rewrote Reset description to eliminate doing Reset wtih Other or Unknown. Added value "NotReseting" to the enumeration to reflect the normal quiesent state that would be Read from this variable.
15. Checked all enumerations and Integer32's to see if Other or Unknown make sense and to see that the minimum value for all Integer32's is specified. The following guidelines were used in this process:
    a. Units may not be either Unknown or Other. They must have a specific value. The values they apply to, however, may be unknown or other. This was done to insure when values are specified, the associated units are always known.
    b. Values that are Read-Write are normally allowed to have Unknown (as well as Other) because these are the values that may not be sensed by the Printer and, therefore, must be written (by some Mgt Appl) if they are to have any known value.
    c. Categories of Read-Write Values

       Dimensions (both in Feed and Cross Feed Directions) & Units

       Capacities and Current Level/Remaining Capacity & Units
    d. The exception on Read-Write and Unknown is within the Optional Groups where it was felt that these groups would be included only for sophisticated devices which would be able to sense the relevant values and would not need Unknown or to write the value.
    e. Values that are Read-only are not allowed a value of Unknown because these should be known to the vendor of the printer. In some cases, however, where the is some doubt about whether a value will always be known, Unknown is allowed.

16. The descriptions of Dimension Units and the objects that report dimensions and addressability were rewritten for consistencey among groups (Input, Output, Marker, Interpreter, MediaPath) and (hopefully) a better definition of intent. This includes applying 15. above.
17. As for 16, Capacities and remaining level/remaining capacity were harmonized and brought into conformity with 15. This affects Input, Output and MarkerSupplies.
18. Strings that were 31 octets long in the last draft became 63 octets in this draft in accordance with issue 42).
19. The YesNoOnOff enum was split apart into the YesNo enum, used only once for TwoWay boolean in the InterpreterTable, and the PresentOnOff enum which is used to indicate whether a feature is present and if so, whether it is On or Off.
20. The MediaType and MediaColor string description have been modified to indicate what the ISO standard values are, but to insure that implementors (and others) can define other valid strings.
21. The rows of the tables have been reordered so that the FeedDim always comes before the XFeedDIm and some consistency of naming has been enforce. In particular, addressability is "Addressability(X)FeedDim". Changed "Across" to "X" to agree with usage elsewhere.
22. The descriptions of Addressability and Margins have been changed to bring them in line with the Working Group minutes.
23. The stacking order enumeration values have been renamed to "FirstToLast" and "LastToFirst" and thedescription has been changed to avoid any reference to how the input arrived at the printer. The description is now expressed solely in terms of how output pages are processed.
24. Added 40 = TIFF and 41 = Diagnostic to the Interpreter Type enumeration; remove marketting statements from enum's 35 and 36.

### 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.

### 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

### 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.

### 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.

### 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”.

## Index


:, 28
AddressabilityFeedDir
in prtMarker, 25
AddressabilityFeedDir in prtInterpreter, 31
AddressabilityUnit
in prtMarker, 25
AddressabilityXFeedDir
in prtMarker, 25
AddressabilityXFeedDir in prtInterpreter, 31
AlertCode
in prtAlert, 35
Bursting
in prtOutput, 24
CapacityUnit
in prtInput, 18
in prtOutput, 22
CharacterSet, 16
in prtGeneral, 16
Class
in prtMarkerSupplies, 26
Color
in prtConsole, 33
ColorantIndex
in prtMarkerSupplies, 26
ConfigChanges
in prtGeneral, 15
ConsoleLocDescriptionalization
in prtConsole, 33
CounterUnit
in prtMarker, 25
Country
in prtGeneral, 15
Cover, 15
CurrentJobCntlLangIndex
in prtChannel, 30
CurrentLevel
in prtInput, 18
CurrentLocatlization
in prtGeneral, 15
Decollating
in prtOutput, 24
DefaultCharSetIn
in prtInterpreter, 31
DefaultCharSetOut
in prtInterpreter, 32
DefaultIndex
in prtInput, 17
in prtInterpreter, 31
in prtMarker, 25
in prtMediaPath, 28
DefaultOrientation
in prtGeneral, 15
DefaultPageDescLangIndex
in prtChannel, 30
Description
in prtAlert, 35
in prtGeneral, 15
in prtInput, 19
in prtInterpreter, 31
in prtMarkerSupplies, 26
in prtMediaPath, 28
in prtOutput, 23
DimUnit
in prtInput, 17
in prtOutput, 23
DisplayBufferIndex
in prtConsole, 33
EastMargin
in prtMarker, 26
Group
in prtAlert, 35
GroupIndex
in prtAlert, 35
ifIndex
in prtChannel, 30
Index
in prtAlert, 34
in prtChannel, 30
in prtConsole, 33
in prtInput, 17
in prtInterpreter, 31
in prtMarker, 25
in prtMarkerColorant, 26
in prtMarkerSupplies, 26
in prtMediaPath, 28
in prtOutput, 22
JobCollated
in prtOutput, 24
LangFamily
in prtInterpreter, 31
LangLevel
in prtInterpreter, 31
Language
in prtGeneral, 15
LangVersion
in prtInterpreter, 31
Level
in prtMarkerSupplies, 26
LifeCount
in prtMarker, 25
LineText
in prtConsole, 33
Localization, 15
in prtConsole, 33
LocalizationCharacterSetIndex
in prtGeneral, 15
Location
in prtAlert, 35
Lock
in prtConsole, 33
MarkerIndex
in prtMarkerColorant, 27
in prtMarkerSupplies, 26
MarkTech
in prtMarker, 25
MaxCapacity
in prtInput, 18
in prtMarkerSupplies, 26
in prtOutput, 22
MaxDimFeedDir
in prtOutput, 23
MaxDimXFeedDir
in prtOutput, 23
MaxMediaFeedDir
in prtMediaPath, 28
MaxMediaXFeedDir
in prtMediaPath, 28
MaxSpeed
in prtMediaPath, 28
MaxSpeedPrintUnit
in prtMediaPath, 28
MediaColor
in prtInput, 20
MediaDimFeedDir
in prtInput, 17
MediaDimXFeedDir
in prtInput, 17, 18
MediaFormParts
in prtInput, 21
MediaName
in prtInput, 18
MediaSizeUnit, 28
MediaType
in prtInput, 20
MediaWeight
in prtInput, 20
MinDimFeedDir, 23
MinDimXFeedDir
in prtOutput, 23
MinMediaFeedDir
in prtMediaPath, 28
MinMediaXFeedDir
in prtMediaPath, 28
Model
in prtInput, 18
in prtOutput, 23
Name
in prtInput, 18
in prtOutput, 23
NorthMargin
in prtMarker, 25
NumberOfDisplayChars
in prtConsole, 33
NumberOfDisplayLines
in prtConsole, 33
OffsetStacking
in prtOutput, 24

OffTime
in prtConsole, 33
OnTime
in prtConsole, 33
PageCollated
in prtOutput, 24
PageDeliveryOrientation
in prtOutput, 24
PowerOnCount
in prtMarker, 25
ProcessColorants
in prtMarker, 25
ProtocolVersion
in prtInterpreter, 30
prtAlertAlertCode, 35
prtAlertDescription, 35
prtAlertGroup, 35
prtAlertGroupIndex, 35
prtAlertIndex, 34
prtAlertLocation, 35
prtAlertSeverityLevel, 34
prtAlertTrainingLevel, 35
prtChannelCurrentJobCntlLangIndex, 30
prtChannelDefaultPageDescLangIndex, 30
prtChannelifIndex, 30
prtChannelIndex, 30
prtChannelState, 30
prtChannelStatus, 30
prtChannelType, 30
prtConsoleColor, 33
prtConsoleDescription, 33
prtConsoleDisplayBufferIndex, 33
prtConsoleLightsIndex, 33
prtConsoleLineText, 33
prtConsoleLocalization, 33
prtConsoleLock, 33
prtConsoleNumberOfDisplayChars, 33
prtConsoleNumberOfDisplayLines, 33
prtConsoleOffTime, 33
prtConsoleOnTime, 33
prtGeneralCharacterSet, 16
prtGeneralCharacterSetIndex, 16
prtGeneralConfigChanges, 15
prtGeneralCountry, 15
prtGeneralCoverIndex, 15
prtGeneralCurrentLocalization, 15
prtGeneralDefaultOrientation, 15
prtGeneralDescription, 15
prtGeneralLanguage, 15
prtGeneralLocalizationCharacterSetIndex, 15
prtGeneralLocalizationIndex, 15
prtGeneralReset, 16
prtGeneralStatus, 15
prtInputCapacityUnit, 18
prtInputCurrentLevel, 18
prtInputDefaultIndex, 17
prtInputDescription, 19
prtInputDimUnit, 17
prtInputIndex, 17
prtInputMaxCapacity, 18
prtInputMediaColor, 20
prtInputMediaDimFeedDir, 17
prtInputMediaDimXFeedDir, 17, 18
prtInputMediaFormParts, 21
prtInputMediaName, 18
prtInputMediaType, 20
prtInputMediaWeight, 20
prtInputModel, 18
prtInputName, 18
prtInputSecurity, 19
prtInputSerialNumber, 18
prtInputStatus, 18
prtInputType, 17
prtInputVendorName, 18
prtInputVersion, 18
prtInterpreterAddressabilityFeedDir, 31
prtInterpreterAddressabilityXFeedDir, 31
prtInterpreterDefaultCharSetIn, 31
prtInterpreterDefaultCharSetOut, 32
prtInterpreterDefaultIndex, 31
prtInterpreterDescription, 31
prtInterpreterIndex, 31
prtInterpreterLangFamily, 31
prtInterpreterLangLevel, 31
prtInterpreterLangVersion, 31
prtInterpreterProtocolVersion, 30
prtInterpreterTwoWay, 32
prtInterpreterVersion, 31
prtMarkerAddressabilityFeedDir, 25
prtMarkerAddressabilityUnit, 25
prtMarkerAddressabilityXFeedDir, 25
prtMarkerColorantIndex, 26
prtMarkerColorantMarkerIndex, 27
prtMarkerColorantRole, 27
prtMarkerColorantTonality, 27
prtMarkerColorantValue, 27
prtMarkerCounterUnit, 25
prtMarkerDefaultIndex, 25
prtMarkerEastMargin, 26
prtMarkerIndex, 25
prtMarkerLifeCount, 25
prtMarkerMarkTech, 25
prtMarkerNorthMargin, 25
prtMarkerPowerOnCount, 25
prtMarkerProcessColorants, 25
prtMarkerSouthMargin, 26
prtMarkerSpotColorants, 25
prtMarkerStatus, 26
prtMarkerSuppliesClass, 26
prtMarkerSuppliesColorantIndex, 26
prtMarkerSuppliesDescription, 26
prtMarkerSuppliesIndex, 26
prtMarkerSuppliesLevel, 26
prtMarkerSuppliesMarkerIndex, 26
prtMarkerSuppliesMaxCapacity, 26
prtMarkerSuppliesSupplyUnit, 26
prtMarkerSuppliesType, 26
prtMarkerWestMargin, 26
prtMediaPathDefaultIndex, 28
prtMediaPathDescription, 28
prtMediaPathIndex, 28
prtMediaPathMaxMediaFeedDir, 28
prtMediaPathMaxMediaXFeedDir, 28
prtMediaPathMaxSpeed, 28
prtMediaPathMaxSpeedPrintUnit, 28
prtMediaPathMaxSpeedTimeUnit, 28
prtMediaPathMediaSizeUnit, 28
prtMediaPathMinMediaFeedDir, 28
prtMediaPathMinMediaXFeedDir, 28
prtMediaPathStatus, 28
prtMediaPathType, 28
prtOutputBursting, 24
prtOutputCapacityUnit, 22
prtOutputDecollating, 24
prtOutputDescription, 23
prtOutputDimUnit, 23
prtOutputIndex, 22
prtOutputJobCollated, 24
prtOutputMaxCapacity, 22
prtOutputMaxDimFeedDir, 23
prtOutputMaxDimXFeedDir, 23
prtOutputMinDimFeedDir, 23
prtOutputMinDimXFeedDir, 23
prtOutputModel, 23
prtOutputName, 23
prtOutputOffsetStacking, 24
prtOutputPageCollated, 24
prtOutputPageDeliveryOrientation, 24
prtOutputRemainingCapacity, 22
prtOutputSecurity, 23

prtOutputSerialNumber, 23
prtOutputSlipsheet, 24
prtOutputStackingOrder, 24
prtOutputStatus, 22
prtOutputType, 22
prtOutputVendorName, 23
prtOutputVersion, 23
RemainingCapacity
in prtOutput, 22
Reset
in prtGeneral, 16
Role
in prtMarkerColorant, 27
Security
in prtInput, 19
in prtOutput, 23
SerialNumber
in prtInput, 18
in prtOutput, 23
SeverityLevel
in prtAlert, 34
Slipsheet
in prtOutput, 24
SouthMargin
in prtMarker, 26
SpotColorants
in prtMarker, 25
StackingOrder
in prtOutput, 24
State
in prtChannel, 30
Status
in prtChannel, 30
in prtGeneral, 15
in prtInput, 18
in prtMarker, 26
in prtMediaPath, 28
in prtOutput, 22
SupplyUnit
in prtMarkerSupplies, 26
Tonality
in prtMarkerColorant, 27
TrainingLevel
in prtAlert, 35
TwoWay
in prtInterpreter, 32
Type
in prtChannel, 30
in prtInput, 17
in prtMarkerSupplies, 26
in prtMediaPath, 28
in prtOutput, 22
Value
in prtMarkerColorant, 27
VendorName
in prtInput, 18
in prtOutput, 23
Version
in prtInput, 18
in prtInterpreter, 31
in prtOutput, 23
WestMargin
in prtMarker, 26