

Printer MIB

Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

To learn the current status of any Internet-Draft, please check the "lid-abstracts-txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

Abstract

This document provides definitions of models and manageable objects for printing environments. The objects included in this MIB apply to physical, as well as logical entities within a printing device. This MIB definition makes explicit references to the Host Resources MIB (RFC 1514), as well as the Interfaces Group of MIB-II (RFC 1213).

Table of Contents

1. Introduction	4
1.1 Network Printing Environment	4
1.2 Printer Device Overview	5
1.3 Categories of Printer Information	6
1.3.1 Descriptions	6
1.3.2 Status	6
1.3.3 Alerts	6
2. Printer Model	7
2.1 Overview of the Printer Model	9
2.2 Printer Sub-Units	9
2.2.1 General Printer	9
2.2.2 Inputs	10
2.2.3 Media	10
2.2.4 Outputs	10
2.2.5 Finishers	10
2.2.6 Markers	11
2.2.7 Media Paths	11
2.2.8 System Controller	11
2.2.9 Interfaces	12
2.2.10 Print Job Delivery Channels	12
2.2.11 Interpreters	12
2.2.12 Console	12
2.2.13 Alerts	13
2.2.13.1 Status and Alerts	13
2.2.13.2 Overall Printer Status	13
2.2.13.2.1 Host MIB Printer Status	15
2.2.13.2.2 Sub-unit Status	17

2.2.13.3 Alert Tables	18
2.2.13.4 Alert Table Management	19
2.3 Read-Write Objects	19
2.4 Enumerations	19
2.4.1 Registering Additional Enumerated Values	19
3. Objects from other MIB Specifications	20
3.1 System Group objects	20
3.2 System Controller	20
3.3 Interface Group objects	20
4. Textual Conventions	21
5. The General Printer Group	39
5.1 The Cover Table	41
5.2 The Localization Table	41
5.3 The System Resources Tables	43
6. The Responsible Party group	44
7. The Input Group	45
8. The Extended Input Group	50
9. The Input Switching Group	52
10. The Output Group	53
11. The Extended Output Group	55
12. The Output Dimensions Group	56
13. The Output Features Group	57
14. The Marker Group	58
15. The Marker Supplies Group	62
16. The Marker Colorant Group	64
17. The Media Path Group	64
18. The Print Job Delivery Channel Group	68
19. The Interpreter Group	71

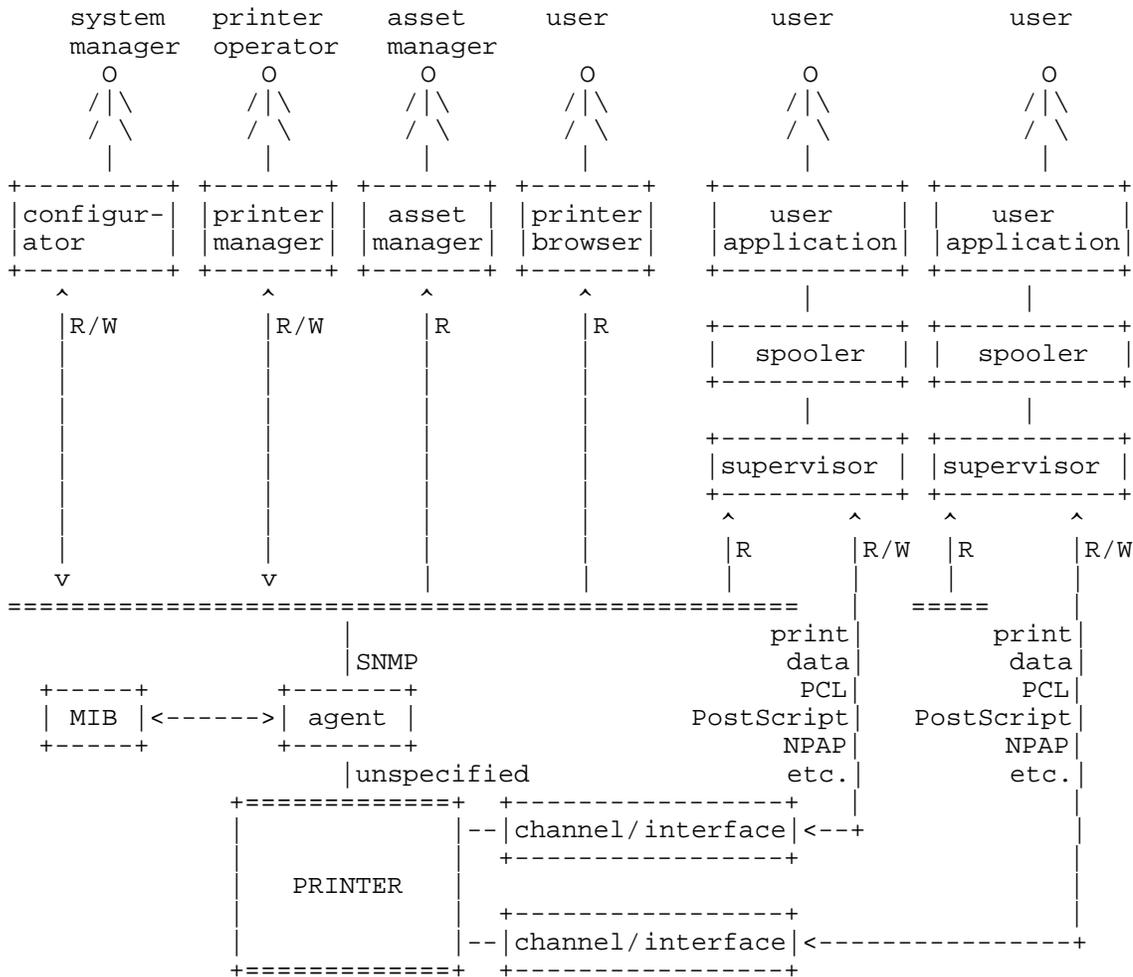
20. The Console Group	74
20.1 The Display Buffer Table	75
20.2 The Console Light Table	76
21. The Alerts Group	78
21.1 The Alert Time Group	81
22. Appendix A - Glossary of Terms	85
23. Appendix B - Media Size Names	87
24. Appendix C - Media Names	89
25. Appendix D - Roles of Users	92
26. Appendix E - Participants	94
27. Security Considerations	95
28. Authors' Addresses	95

1. Introduction

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 overlap highly 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, reconfigured 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. 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.

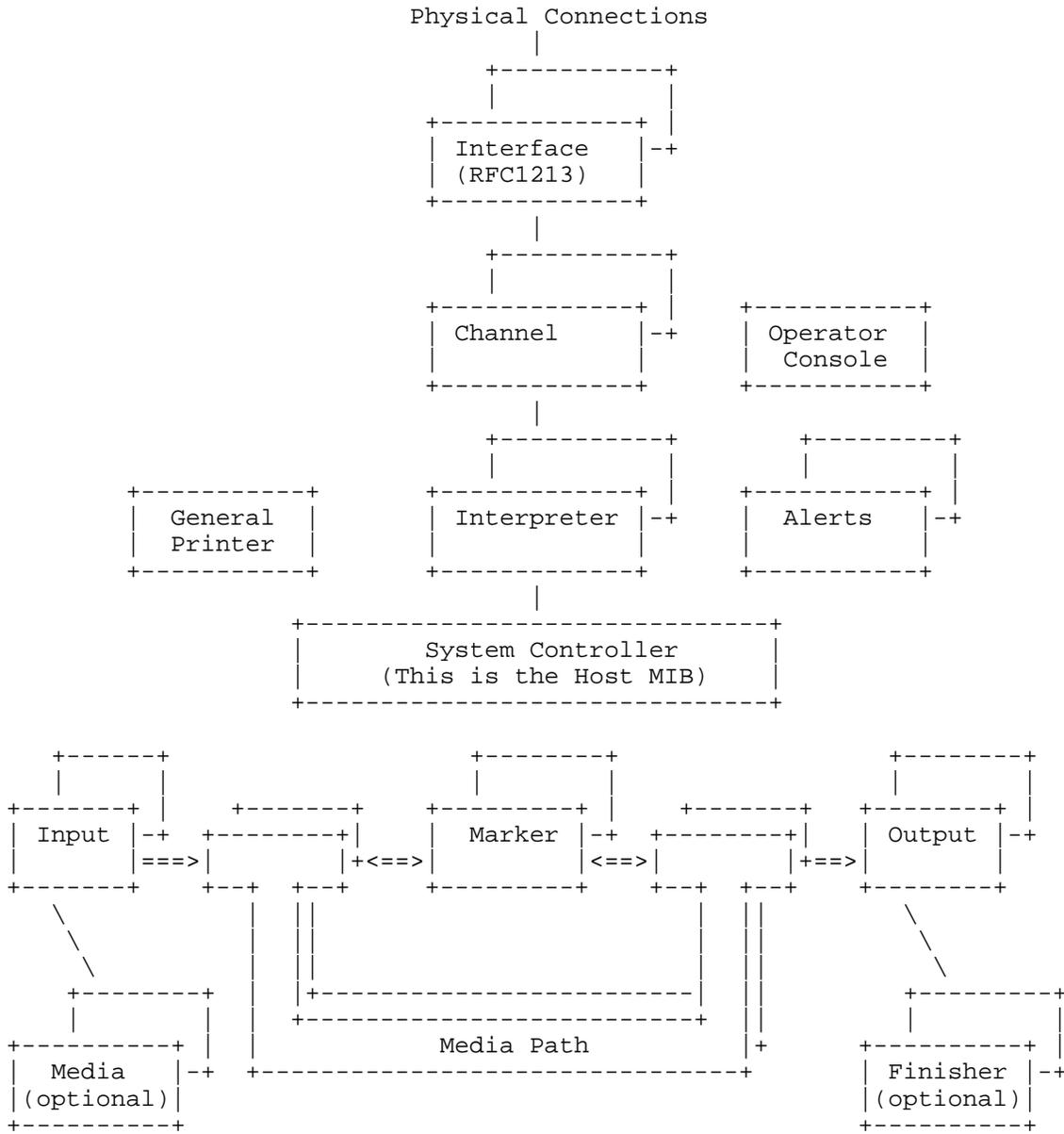
Description, status and alert information about the printer can be thought of as a data base describing the printer. The management application for a printer will want to view the printer data base differently depending on how and for what purposes the information in the data base is needed.

2. Printer Model

In order to accomplish 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 covers. 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.

Localization is only performed on those strings in the MIB that are explicitly marked as being localized. All other character strings are returned in ASCII.

The character set portion of the general printer sub-unit is responsible for identifying the possible character sets that are used by the interpreters, the operator console, and in network 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 are 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 operations 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. Finishers are not described in this MIB.

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

2.2.6. Markers

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

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

2.2.7. Media Paths

The media paths encompass the mechanisms in the printer that move the media through the printer and connect all other media related sub-units: inputs, outputs, markers and finishers. A printer contains one or more media paths. These are represented by the Media Path Group in the model. The Media Path group has some objects 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 are 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 (RFC 1213). 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. Print Job Delivery Channels

The print job delivery 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 Print Job Delivery 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. Print Job Delivery Channel sub-units can, and usually 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 exception to this rule is when the "alert alert" trap is generated. The alert sub-unit is represented by the Alerts Group and, in particular, the Alert Table. This table contains information on the severity, sub-unit, detailed location within the sub-unit, alert code and description of each critical alert that is currently active within the printer. Each reportable event causes an entry to be made in the Alert Table.

2.2.13.1. Status and Alerts

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

2.2.13.2. Overall Printer Status

Of the many states a printer can be in, certain states are more "interesting" because of the distinct actions they are likely to provoke in the administrator. These states may be applied to the printer as a whole, or to a particular sub-unit of the printer. These named states are:

Non Critical Alert Active - For the printer this means that one or more sub-units have a non-critical alert active. For a sub-unit, this means that the sub-unit has a non-critical alert active.

Critical Alert Active - For the printer this means that one or more sub-units have a critical alert active. For a sub-unit, this means that the sub-unit has a critical alert active.

Unavailable - The printer or sub-unit is unavailable for use (this is the same as "broken" or "down" in other terminologies). A trained service person is typically necessary to make it available. **Busy / Temporarily Unavailable** - The printer or 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-line or off-line - The printer 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.

Standby - The printer or sub-unit is unavailable for use because it is partially powered down and may need some period of time to become fully operational again. A unit in Standby state shall respond to network management requests.

Available - The printer or subunit is functioning normally.

Idle - The printer or subunit is not processing any print jobs

Active - The printer or subunit is working on one or more print jobs

Busy - The printer or subunit is "Active" and has also reached a resource threshold whereby it will not accept any new print jobs for a finite period of time. The expected next state transition from the "busy" state is to the "active" state, implying that resources are again available to accept new print jobs.

The Host Resources MIB (RFC 1514) provides three status objects that can be used to describe the status of a printer: (1) hrDeviceStatus in the entry in the hrDeviceTable; (2) hrPrinterStatus in the hrPrinterTable; and (3) hrPrinterDetectedErrorState in the hrPrinterTable. These objects describe many of the states that a printer can be in. The following table shows how the states named above can be recognized by inspecting the values of the three printer-related objects 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, lowToner, or serviceRequested
Critical Alert Active	down(5)	other(1)	could be: jammed, noPaper, noToner, coverOpen, or 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)	
Standby	running(2)	other(1)	

These named states are only a subset of the possible states - they are not an exhaustive list of the possible states. Nevertheless, several things should be noted. When using these states, it is not possible to detect when both critical and non-critical alerts are pending - if both are pending, the Critical Alert Active state will prevail. In addition, a printer in the Standby state will be represented in the Host MIB with a device status of running(2) and a printer status of other(1), a set of states that don't uniquely distinguish this important printer state.

Although the above mapping is workable, it would be improved with a few additions to hrDeviceStatus and hrPrinterStatus in the Host Resources MIB. In particular, it would be appropriate to add a "standby" enumeration to hrDeviceStatus. Similarly, it would be useful to add the following states to hrPrinterStatus:

"offline" to indicate that reason for the printer being down (instead of having to use "other") which allows both "warning" and "offline" to indicate going offline and "down" and "offline" to indicate offline and "notApplicable" to cover cases, such as "standby", where the device state completely describes the state of the device.

Detailed status per sub-unit is reported in the sub-unit status fields.

2.2.13.2.1. Host MIB Printer Status

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

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

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

::= { hrPrinterEntry 1 }

hrPrinterDetectedErrorState OBJECT-TYPE

SYNTAX OCTET STRING

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This object represents any error conditions detected by the printer. The error conditions are encoded as bits in an octet string, with the following definitions:

Condition	Bit #	hrDeviceStatus
lowPaper	0	warning(3)
noPaper	1	down(5)
lowToner	2	warning(3)
noToner	3	down(5)
coverOpen	4	down(5)
jammed	5	down(5)
offline	6	down(5)
serviceRequested	7	warning(3)

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

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

This object is useful for alerting an operator to specific warning or error conditions that may occur, especially those requiring human intervention."

::= { hrPrinterEntry 2 }

2.2.13.2.2. Sub-unit Status

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

The SubUnitStatus is an integer that is the sum of 5 distinct values, Availability, Non-Critical, Critical, On-line, and Transitioning. These values are:

Availability	value
Available and Idle	0 000'b
Available and Standby	2 010'b

Available and Active	4	100'b
Available and Busy	6	110'b
Unavailable and OnRequest	1	001'b
Unavailable because Broken	3	011'b
Unknown	5	101'b

Non-Critical

No Non-Critical Alerts	0
Non-Critical Alerts	8

Critical

No Critical Alerts	0
Critical Alerts	16

On-Line

Intended state is On-Line	0
Intended state is Off-Line	32

Transitioning

At intended state	0
Transitioning to intended state	64

For example, an input (tray) that jammed on the next to the last page may show a status of 27 (unavailable because broken (3) + a critical state (16), jammed, and a noncritical state (8), low paper).

2.2.13.3. Alert Tables

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

The Alert Table contains some information that is redundant, for example that an event has occurred, and some information that is only represented in the Alert Table, for example the additional information. A single table was used because a single entry in a group could cause more than one alert, for example paper jams in more than one place in a media path. Associating the additional information with the entry in the affected group would only allow one report where associating the additional information with the alert makes multiple reports possible. Every time an alert occurs in the printer, the printer makes one or more entries into the Alert Table. The printer determines if an event is to be classified as critical or non-critical. If the severity of the Alert is "critical", the printer sends a trap or event notification to the host indicating that the table has changed. Whether or not a trap is sent, the management application is expected to poll the printer on a regular basis and to read and parse the table to determine what conditions have changed, in order to provide reliable information to the management application user.

2.2.13.4. Alert Table Management

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

There are basically two kinds of events that produce alerts: binary change events and unary 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 that 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 signaled 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. Removing the leading edge entry may cause the unary change event "alertRemovalofBinaryChangeEvent" to be added to the table. With binary change 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 unary 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. If 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 non-critical 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 critical 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 critical 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 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

Some of the objects in the printer MIB 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 objects which hold the descriptive or existence values for printers that cannot sense the values. Thus many of the objects in the MIB are given read-write access, but a printer implementation might only permit a management operation to change the value if the printer could not sense the value itself. Therefore, the ability to change the value of a read-write object may depend on the implementation of the agent. Note that even though some objects explicitly state the behaviour of conditional ability to change values, any read-write object may act that way.

Generally, an object is given read-write access in the Printer MIB specification if:

- 1.The object 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 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, 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 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 either strategy is acceptable.

It is an implementation-specific matter as to whether or not MIB object values are persistent across power cycles or cold starts. It is particularly important that the values of the prtMarkerLifeCount object persist throughout the lifetime of the printer. Therefore, if the value of any MIB object persists across power cycles, then the prtMarkerLifeCount object must also persist.

2.4. Enumerations

Enumerations (enums) are sets of symbolic values defined for use with one or more objects. Some common enumeration sets are assigned a symbolic data type name (textual convention). These enumerations are listed at the beginning of this specification.

2.4.1. 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 "enumeration (n)", where n can be 1, 2 or 3 can be found in the various tables. The definitions of these types of enumerations are:

enumeration (1) All the values are defined in the Printer MIB specification (RFC for the Printer MIB). Additional enumerated values require a new RFC. Type 1 enumerations are typically used where changes to the enumeration are either unlikely or will have a significant impact on the structure of the MIB or implementation of the MIB in management applications.

Some criteria that suggest using a type 1 enumeration are:

a) the set of values in the enumeration is thought to be known, e.g., faceUp and faceDown

b) the enumeration defines a set of units of measure which must be understood by a management application to be able to correctly display the value of an object that measurement unit controls; and

c) the enumeration is tied to the structure of the MIB or the model on which the MIB is based, e.g., the prtAlertGroup enumeration is tied to the OIDs for the related tables.

enumeration (2) An initial set of values are defined in the Printer MIB specification. Additional enumerated values are registered after review by this working group. The initial versions of the MIB will contain the values registered so far. After the MIB is approved, additional values will be registered through IANA after approval by this working group. The current set of approved values should always be obtained from the IANA registry. Type 2 enumerations are typically used where it is important to insure consistent usage of the enumeration values; that is, to insure that the same entity does not get two different enumerations values, or two different entities do not get the same enum value.

enumeration (3) An initial set of values are defined in the Printer MIB specification. Additional enumerated values are registered without working group review. The initial versions of the MIB will contain the values registered so far. After the MIB is approved, additional values will be registered through IANA without approval by this working group. The current set of approved values should always be obtained from the IANA registry. Type 3 enumerations are used for enumerations that can be extended without any controls; an example is the prtMarkerSuppliesType, which can be extended as needed by any manufacturer to describe the supplies required by a new printer.

3. Groups from other MIB Specifications

This section identifies the groups from other MIBs that shall be supported to supplement and complete a printer MIB implementation. The section also describes some of the less obvious characteristics of the Printer MIB structure that are related to the inclusion of these other MIB groups and/or objects.

3.1. System Group

All objects in the system group of MIB-II (RFC 1213) shall be implemented; however, as described in paragraph 3.4, implementers should carefully consider what constitutes the "system".

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 Group objects

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

```
Printer-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    MODULE-IDENTITY, OBJECT-TYPE, experimental, Counter32, Integer32,  
        TimeTicks, NOTIFICATION-TYPE, OBJECT-IDENTITY FROM SNMPv2-SMI  
    TEXTUAL-CONVENTION FROM SNMPv2-TC  
    MODULE-COMPLIANCE, OBJECT-GROUP FROM SNMPv2-CONF  
    hrDeviceIndex, hrStorageIndex FROM HOST-RESOURCES-MIB;
```

```
printmib MODULE-IDENTITY
```

```
    LAST-UPDATED "9411250000Z"  
    ORGANIZATION "IETF Printer MIB Working Group"  
    CONTACT-INFO  
        "Randy Turner  
        Sharp Laboratories of America  
        5750 NW Pacific Rim Blvd  
        Camas, WA 98607  
        rturner@sharplabs.com"  
    DESCRIPTION  
        "The MIB module for management of printers."  
    ::= { mib-2 43 }
```

```
--  
-- Textual conventions for this MIB module  
--  
--  
-- Generic unspecific textual conventions  
--
```

```
PrtMediaUnitTC ::= TEXTUAL-CONVENTION
```

```
    STATUS          current  
    DESCRIPTION  
        "Units of measure for media dimensions."  
    -- This is a type 1 enumeration.  
    SYNTAX          INTEGER {
```

```
        tenThousandthsOfInches(3), -- .0001  
        micrometers(4)  
    }
```

```
PrtCapacityUnitTC ::= TEXTUAL-CONVENTION  
    STATUS      current  
    DESCRIPTION  
        "Units of measure for media capacity."  
    -- This is a type 1 enumeration.  
    SYNTAX      INTEGER {  
        tenThousandthsOfInches(3), -- .0001  
        micrometers(4),  
        sheets(8),  
        feet(16),  
        meters(17)  
    }
```

```
PrtSupplementaryPageContentTC ::= TEXTUAL-CONVENTION  
    STATUS      current  
    DESCRIPTION  
        "Supplementary pages are added by the printing system and are  
        not part of the actual print job. Examples include error and  
        banner pages. This textual convention is used to specify the  
        information content of these pages."  
    SYNTAX      INTEGER {  
        noPage(3),           -- no page will be produced  
        concise(4),         -- content of page is concise  
        verbose(5)         -- content of page is verbose  
    }
```

```
PrtPrintOrientationTC ::= TEXTUAL-CONVENTION  
    -- This value is a type 1 enumeration  
    STATUS      current  
    DESCRIPTION  
        "A generic representation for printing orientation on a "page"."  
    SYNTAX      INTEGER {  
        other(1),  
        portrait(3),  
        landscape(4)  
    }
```

```
PrtCoverStatusTC ::= TEXTUAL-CONVENTION  
    STATUS      current  
    DESCRIPTION  
        "Values for encoding the state of a particular cover or access panel  
        on the printer case or enclosure."  
    SYNTAX      INTEGER {  
        other(1),  
        coverOpen(3),  
        coverClosed(4),  
        interlockOpen(5),  
        interlockClosed(6)  
    }
```

```
PrtSubUnitStatusTC ::= TEXTUAL-CONVENTION  
    STATUS      current  
    DESCRIPTION
```

"Status of a printer sub-unit.

The SubUnitStatus is an integer that is the sum of 5 distinct values, Availability, Non-Critical, Critical, On-line, and Transitioning. These values are:

Availability	value	
Available and Idle	0	000'b
Available and Standby	2	010'b
Available and Active	4	100'b
Available and Busy	6	110'b
Unavailable and OnRequest	1	001'b
Unavailable because Broken	3	011'b
Unknown	5	101'b

Non-Critical

No Non-Critical Alerts	0
Non-Critical Alerts	8

Critical

No Critical Alerts	0
Critical Alerts	16

On-Line

Intended state is On-Line	0
Intended state is Off-Line	32

Transitioning

At intended state	0
Transitioning to intended state	64

"

SYNTAX INTEGER (0..126)

```
PresentOnOff ::= TEXTUAL-CONVENTION
  STATUS      current
  DESCRIPTION
    "Presence and configuration of a device or feature."
  -- This is a type 1 enumeration.
  SYNTAX      INTEGER {
                other(1),
                on(3),
                off(4),
                notPresent(5)
              }
```

```
CodedCharSet ::= TEXTUAL-CONVENTION
  STATUS      current
  DESCRIPTION
    "A coded character set value that specifies both a set of
    characters that may be used and an encoding (as one or more
    octets) that is used to represent the characters in the
    set. These values are to be used to identify the encoding
    employed for strings in the MIB where this is not fixed by
    the MIB."
```

Some objects that allow a choice of coded character set are: the `prtLocalizationCharacterSet` object in the `LocalizationTable` and `prtInterpreterDefaultCharSetIn`. The `prtGeneralCurrentLocalization` and `prtConsoleLocalization` objects in turn contain the index in the `LocalizationTable` of the current localization (country, language, and coded character set) of the 'description' objects and the console, respectively.

The space of the coded character set enumeration has been divided into three regions. The first region (3-999) consists of coded character sets that have been standardized by some standard setting organization. This region is intended for standards that do not have subset implementations. The second region (1000-1999) is for the Unicode and ISO/IEC 10646 coded character sets together with a specification of a (set of) sub-repertoires that may occur. The third region (>1999) is intended for vendor specific coded character sets.

NOTE: Unicode and ISO 10646 character coded data may be processed and stored in either Big Endian (most significant octet first) or Little Endian (least significant octet first) order. Intel x86, VAX, and Alpha/AXP architectures are examples of Little Endian processor architectures. Furthermore, in environments where either order may occur, so-called Unicode BYTE ORDER MARK (BOM) character (which is ISO 10646 ZERO WIDTH NO BREAK SPACE), coded as FEFF in two octets and 0000FEFF in four octets is used at the beginning of the data as a signature to indicate the order of the following data (See ISO 10646 Annex F). Thus either ordering and BOM may occur in print data streams sent to the interpreter. However, ISO 8824/8825 (ASN.1/BER) used by SNMP is quite clear that Big Endian order shall be used and BOM shall NOT be used in transmission in the protocol. Transmitting Unicode in Big Endian order in SNMP should not prove to be a hardship for Little Endian machines, since SNMP ASN.1/BER requires integers to be transmitted in Big Endian order as well. So SNMP implementations on Little Endian machines are already reversing the order of integers to make them Big Endian for transmission via SNMP. Also Unicode characters are usually treated as two-octet integers, not short text strings, so that it will be straightforward for Little Endian machines to reverse the order of Unicode character octets as well before transmitting them and after receiving them via the SNMP protocol.

Where a given coded character set may be known by more than one name, the most commonly known name is used as the name of the enumeration and other names are shown in the comments. The comments also indicate where to find detailed information on the coded character set and briefly characterize its relationship to other similar coded character sets.

The current list of character sets and their enumerated values used to reference them is contained in the IANA Character Set registry. The enum value is indicated by the `MIBenum` entry in the registry. The enum symbol is

indicated by the Alias that starts with 'cs' for character set.

The IANA character sets registry is available via anonymous ftp.

The ftp server is ftp.isi.edu.
The subdirectory is /in-notes/iana/assignments/.

The file name is character-sets.

To add a character set to the IANA Registry:

1. Format an entry like those in the current list, omitting the MIBenum value.
2. Send the entry with a request to add the entry to the character set list to iana@ISI.EDU.
3. The IANA will supply a unique MIBenum value and update the list."

-- This is a type 3 enumeration.

```
SYNTAX      INTEGER {
  other(1)           -- used if the designated coded
                    -- character set is not currently in
                    -- the enumeration
```

-- See IANA Registry for standard character sets in the
-- MIBenum range of 3-999.

-- See IANA Registry for Unicode and vendor-supplied
-- combinations of ISO collections and character sets based
-- on Unicode in the MIBenum range of 1000-1999.
-- See IANA Registry for vendor developed character sets
-- in the MIBenum range of 2000-xxxx.

}

--
-- General Group textual-conventions
--

```
PrtGeneralResetTC ::= TEXTUAL-CONVENTION
  STATUS      current
  DESCRIPTION
    "Values for reading and writing the prtGeneralReset object.
```

This value is a type 3 enumeration.

If a device does not have NVRAM, the device shall none the less respond to a SET with the value resetToNVRAM(5) with some sort of "warm reset" that resets the device to some implementation-defined state that is preferably under control of the system administrator by some means outside the scope of this MIB specification."

```
SYNTAX      INTEGER {
  notResetting(3),
  powerCycleReset(4), -- Cold Start
  resetToNVRAM(5), -- Warm Start
  resetToFactoryDefaults(6) -- Reset contents of
```

```

-- NVRAM to factory
-- defaults
    }

--
-- Channel Group textual-conventions
--

PrtChannelStateTC ::= TEXTUAL-CONVENTION
-- This value is a type 1 enumeration
STATUS    current
DESCRIPTION
    "The state of this print job delivery channel. The value determines
    whether control information and print data is allowed
    through this channel."
SYNTAX    INTEGER {
            other(1),
            printDataAccepted(3),
            noDataAccepted(4)
        }

PrtChannelTypeTC ::= TEXTUAL-CONVENTION
STATUS    current
DESCRIPTION
    "This enumeration indicates the type of channel that is
    receiving jobs.

    This is a type 2 enum."
SYNTAX    INTEGER {
            other(1),
            chSerialPort(3),
            chParallelPort(4),
            chIEEE1284Port(5),
            chSCSIPort(6),
            chAppleTalkPAP(7),      -- AppleTalk Printer Access
                                     -- Protocol (PAP)
            -- prtChannelInformation Keywords:
            --
            -- Printer Name
            --
            -- Keyword:          "Name"
            -- Status:          Optional
            -- Multiplicity     Single
            -- Description:     The name of the printer within
            --                   the AppleTalk naming scope
            chLPDServer(8),        -- RFC 1179
            chNetwareRPrinter(9),  -- Novell, Inc.
            -- prtChannelInformation keywords:
            -- Print Server Name
            -- Keyword:         "PServer"
            -- Syntax:         Name
            -- Status:         Mandatory
            -- Multiplicity:   Single
            -- Description:    The Pserver's SAP name
            --
            -- Printer Number
            -- Keyword:        "Printer"

```

```
-- Syntax:      Integer
-- Status:      Mandatory
-- Multiplicity  Single
-- Description: The printer number
-- There must be one Pserver and one Printer entry
chNetwarePServer(10), -- Novell, Inc.
-- prtChannelInformation keywords:
-- Server Name
-- Keyword:     "Server"
-- Syntax:      Name
-- Status:      Mandatory
-- Multiplicity  Single
-- Description: The name of the server on which
-- the queue resides.
--
-- Queue
--
-- Name:        "Queue"
-- Syntax:      Name
-- Status:      Mandatory
-- Multiplicity  Single
-- Description: The name of the queue on the
-- server
--
-- NDS Tree
--
-- Keyword:     "NDSTree"
-- Syntax:      Name
-- Status:      Mandatory
-- Multiplicity  Multiple
-- Description: The NDS Tree SAP name
--
-- NDS Queue
--
-- Keyword:     "NDSQueue"
-- Syntax:      Name
-- Status:      Mandatory
-- Multiplicity  Single
-- Description: The name of the queue object in
-- the associated NDS Tree
-- There must be exactly one Server and one Queue
-- entry or exactly one NDSTree and one NDSQueue
-- Entry.
chPort9100(11),      -- Bi-directional printing protocol
                    -- utilizing TCP port 9101 for control
                    -- and TCP port 9100 for data.
chAppSocket(12),    -- Adobe Systems, Inc.
chFTP(13),          -- RFC 959
chTFTP(14),        -- RFC 1350
chDLCLLCPort(15),
chIBM3270(16),      -- IBM Coax
chIBM5250(17),     -- IBM Twinax
chFax(18),
chIEEE1394(19),
chTransport1(20),  -- TCP port 35, see reserved
                    -- TCP port list in RFC 1700
                    -- or current "Assigned Numbers"
                    -- RFC. This RFC should also be
                    -- referenced for other channel
```

```

-- enumerations utilizing TCP
-- port numbers 0 through 1024.
chCPAP(21), -- TCP port 170, Digital
-- Equipment Corp.
chPCPrint(26), -- Banyan
chServerMessageBlock(27),
-- File/Print sharing protocol used by
-- various network operating systems
-- from IBM 3Com, Microsoft and others
-- prtChannelInformation Keywords:
--
-- Service Name
--
-- Keyword: "Name"
-- Status: Optional
-- Multiplicity: Single
-- Description: The service name of the printer

chPSM(28), -- Printing Systems Manager, IBM
chSystemObjectManager(31), -- IBM
chDECLAT(32), -- Digital Equipment Corp.
chNPAP(33),
chUSB(34), -- Universal Serial Bus
chIRDA(35), -- Infrared Data Assoc. Prot.
chPrintXChange(36), -- PrintXChange Protocol
chPortTCP(37), -- A unidirectional "raw" TCP
-- channel that uses an
-- administratively assigned TCP
-- port address.
chBidirPortTCP(38), -- A bidirectional version of chPortTCP
chUNPP(39), -- Universal Network Printing Protocol
-- (UNPP). A bi-directional, multiport
-- network printing application
-- protocol available on multiple
-- transport protocols.
-- Underscore, Inc.
-- Contact: info@underscore.com
chAppleTalkADSP(40), -- AppleTalk Data Stream
-- Protocol.
-- ADSP is part of the AppleTalk
-- suite of protocols.
-- It is a symmetric, connection-oriented
-- protocol that makes possible
-- the establishment and maintenance of
-- full-duplex streams of data bytes
-- between two sockets in an AppleTalk
-- internet.
-- See Inside AppleTalk, second Edition, by
-- Sidhu, Andrews and Oppenheimer.
chPortSPX(41), -- Sequenced Packet Exchange (SPX) socket.
-- Novell, Inc. Similar to TCP, a
-- bi-directional data pipe using Novell
-- SPX as a transport.
--
-- prtChannelInformation keywords:
--
-- Network Number
--
-- Keyword: "Net"
-- Syntax: HexString
```

```
-- Status:          Mandatory
-- Multiplicity Single
-- Description: The network number
--
-- Node Number
--
-- Keyword:         "Node"
-- Syntax:          HexString
-- Status:          Mandatory
-- Multiplicity Single
-- Description: The Node Number
--
-- Socket Number
--
-- Keyword:         "Socket"
-- Syntax:          HexString
-- Status:          Mandatory
-- Multiplicity Single
-- Description: The SPX socket number
-- There must be exactly one "Net" and one "Node"
-- and one "Socket" entry. A HexString is a binary
-- value represented as a string of ASCII
-- characters using hexadecimal notation.
chPortHTTP(42)  -- Hypertext Transfer Protocol. See IETF
                -- documents relating to HTTP 1.0/1.1
                -- (RFCs XXXX,etc.)
chNDPS(43)     -- Novell, Inc.
                -- prtChannelInformation keywords:
--
-- Printer Agent Name
--
-- Keyword:         "PA"
-- Syntax:          Name
-- Status:          Mandatory
-- Multiplicity Single
-- Description: The NDPS Printer Agent Name
}
```

-- Interpreter Group textual conventions

```
PrtInterpreterLangFamilyTC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "This enumeration indicates the type of interpreter that is
        receiving jobs.

        This value is a type 2 enumeration."
    SYNTAX      INTEGER {
        other(1),
        langPCL(3),      -- PCL. Starting with PCL version 5,
                        -- HP-GL/2 is included as part of the
                        -- PCL language.
                        -- PCL and HP-GL/2 are registered
                        -- trademarks of Hewlett-Packard Company.
        langHPGL(4),    -- Hewlett-Packard Graphics Language.
                        -- HP-GL is a registered trademark of
                        -- Hewlett-Packard Company.
        langPJL(5),    -- Peripheral Job Language. Appears in the
                        -- data stream between data intended for a
```

```

-- page description language.
-- Hewlett-Packard Co.
langPS(6), -- PostScript Language (tm)
-- Postscript - a trademark of Adobe
-- Systems Incorporated which may be
langIPDS(7), -- registered in certain jurisdictions
-- Intelligent Printer Data Stream
-- Bi-directional print data stream for
-- documents consisting of data objects
-- (text, image, graphics, bar codes),
-- resources (fonts, overlays) and page,
-- form and finishing instructions.
-- Facilitates system level device
-- control, document tracking and error
-- recovery throughout the print process.
-- Pennant Systems, IBM
langPPDS(8), -- IBM Personal Printer Data Stream.
-- Originally called IBM ASCII, the name
-- was changed to PPDS when the Laser
-- Printer was introduced in 1989.
-- Lexmark International, Inc.
langEscapeP(9), -- Epson Corp.
langEpson(10),
langDDIF(11), -- Digital Document Interchange Format
-- Digital Equipment Corp., Maynard MA
langInterpress(12), -- Xerox Corp.
langISO6429(13), -- ISO 6429. Control functions for Coded
-- Character Sets (has ASCII control
-- characters, plus additional controls for
-- character imaging devices.)
-- ISO Standard, Geneva, Switzerland
langLineData(14), -- line-data: Lines of data as separate
-- ASCII or EBCDIC records and containing
-- no control functions (no CR, LF, HT, FF,
-- etc.). For use with traditional line
-- printers. May use CR and/or LF to
-- delimit lines, instead of records. See
-- ISO 10175 Document Printing Application
-- (DPA)
-- ISO standard, Geneva, Switzerland
langMODCA(15), -- Mixed Object Document Content
-- Architecture
-- Definitions that allow the composition,
-- interchange, and presentation of final
-- form documents as a collection of data
-- objects (text, image, graphics, bar
-- codes), resources (fonts, overlays) and
-- page, form and finishing instructions.
-- Pennant Systems, IBM
langREGIS(16), -- Remote Graphics Instruction Set,
-- Digital Equipment Corp., Maynard MA
langSCS(17), -- SNA Character String
-- Bi-directional print data stream for SNA
-- LU-1 mode of communications
-- IBM
langSPDL(18), -- ISO 10180 Standard Page Description
-- Language
-- ISO Standard
langTEK4014(19), -- Tektronix Corp.
langPDS(20),
```

```
langIGP(21),      -- Printronix Corp.
langCodeV(22),   -- Magnum Code-V, Image and printer control
                 -- language used to control impact/dot-
                 -- matrix printers.
                 -- QMS, Inc., Mobile AL
langDSCDSE(23),  -- DSC-DSE: Data Stream Compatible and
                 -- Emulation Bi-directional print data
                 -- stream for non-SNA (DSC) and SNA LU-3
                 -- 3270 controller (DSE) communications
                 -- IBM
langWPS(24),     -- Windows Printing System, Resource based
                 -- command/data stream used by Microsoft At
                 -- Work Peripherals.
                 -- Developed by the Microsoft Corporation.
langLN03(25),   -- Early DEC-PPL3, Digital Equipment Corp.
langCCITT(26),  --
langQUIC(27),   -- QUIC (Quality Information Code), Page
                 -- Description Language for laser printers.
                 -- Included graphics, printer control
                 -- capability and emulation of other well-
                 -- known printer .
                 -- QMS, Inc.
langCPAP(28),   -- Common Printer Access Protocol
                 -- Digital Equipment Corp.
langDecPPL(29), -- Digital ANSI-Compliant Printing Protocol
                 -- (DEC-PPL)
                 -- Digital Equipment Corp.
langSimpleText(30), -- simple-text: character coded data,
                 -- including NUL, CR , LF, HT, and FF
                 -- control characters. See ISO 10175
                 -- Document Printing Application (DPA)
                 -- ISO standard, Geneva, Switzerland
langNPAP(31),   -- Network Printer Alliance Protocol (NPAP).
                 -- This protocol has been superseded
                 -- by the IEEE 1284.1 TIPSII standard.
                 -- (ref. LangTIPSII(49)).
langDOC(32),    -- Document Option Commands, Appears in the
                 -- data stream between data intended for a
                 -- page description .
                 -- QMS, Inc.
langimPress(33), -- imPRESS, Page description language
                 -- originally developed for the ImageServer
                 -- line of systems. A binary language
                 -- providing representations for text,
                 -- simple graphics (rules, lines, conic
                 -- sections), and some large forms (simple
                 -- bit-map and CCITT group 3/4 encoded).The
                 -- language was intended to be sent over an
                 -- 8-bit channel and supported early
                 -- document preparation languages (e.g. TeX
                 -- and TROFF).
                 -- QMS, Inc.
langPinwriter(34), -- 24 wire dot matrix printer for
                 -- USA, Europe, and Asia except Japan.
                 -- More widely used in Germany, and some
                 -- Asian countries than in US.
                 -- NEC
langNPDL(35),   -- Page printer for Japanese
                 -- market.
                 -- NEC
```

```
langNEC201PL(36), -- Serial printer language used in the
                  -- Japanese market.
                  -- NEC
langAutomatic(37), -- 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.
langPages(38), -- Page printer Advanced Graphic Escape Set
               -- IBM Japan
langLIPS(39), -- LBP Image Processing System
langTIFF(40), -- Tagged Image File Format (Aldus)
langDiagnostic(41), -- A hex dump of the input to the
                   -- interpreter
langPSPrinter(42), -- The PostScript Language used for
                   -- control (with any PDLs)
                   -- Adobe Systems Incorporated
langCaPSL(43), -- Canon Print Systems Language
langEXCL(44), -- Extended Command Language
               -- Talaris Systems Inc.
langLCDS(45), -- Line Conditioned Data Stream
               -- Xerox Corporation
langXES(46), -- Xerox Escape Sequences
               -- Xerox Corporation
langPCLXL(47), -- Printer Control Language. Extended
               -- language features for printing, and
               -- printer control. Technical reference
               -- manual # TBD.
               -- Hewlett-Packard Co.
langART(48), -- Advanced Rendering Tools (ART). Page
              -- Description language originally
              -- developed for the Laser Press printers.
              -- Technical reference manual: "ART IV
              -- Reference Manual", No F33M.
              -- Fuji Xerox Co., Ltd.
langTIPSI(49), -- Transport Independent Printer System
               -- Interface (ref. IEEE Std. 1284.1)
langPrescribe(50), -- Page description and printer control
                  -- language. It can be described with
                  -- ordinary ASCII characters.
                  -- Technical reference manual:
                  -- "PRESCRIBE II Programming Manual"
langLinePrinter(51), -- A simple-text character stream which
                    -- supports the control codes LF, VT,
                    -- FF and CR plus Centronics or
                    -- Dataproducts Vertical Format
                    -- Unit (VFU).
                    -- language is commonly used on many
                    -- older model line and matrix printers.
langIDP(52), -- Imaging Device Protocol
              -- Apple Computer.
LangXJCL(53) -- Xerox Corp.
}

```

```
--
-- Input/Output Group Textual Conventions
--
```

```
PrtInputTypeTC ::= TEXTUAL-CONVENTION
  STATUS current
  DESCRIPTION
    "The type of technology (discriminated primarily according to feeder
    mechanism type) employed by a specific component or components."
  SYNTAX INTEGER {
    other(1),
    unknown(2),
    sheetFeedAutoRemovableTray(3),
    sheetFeedAutoNonRemovableTray(4),
    sheetFeedManual(5),
    continuousRoll(6),
    continuousFanFold(7)
  }
```

```
PrtOutputTypeTC ::= TEXTUAL-CONVENTION
-- This is a type 2 enumeration
  STATUS current
  DESCRIPTION
    "The Type of technology supported by this output sub-unit"
  SYNTAX INTEGER {
    other(1),
    unknown(2),
    removableBin(3),
    unRemovableBin(4),
    continuousRollDevice(5),
    mailBox(6),
    continuousFanFold(7)
  }
```

```
PrtOutputStackingOrderTC ::= TEXTUAL-CONVENTION
-- This is a type 1 enumeration
  STATUS current
  DESCRIPTION
    "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."
  SYNTAX INTEGER {
    unknown(2),
    firstToLast(3),
    lastToFirst(4)
  }
```

```
PrtOutputPageDeliveryOrientationTC ::= TEXTUAL-CONVENTION
-- This is a type 1 enumeration
  STATUS current
  DESCRIPTION
    "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."
  SYNTAX INTEGER {
    faceUp(3),
    faceDown(4)
  }
```

}

--
-- Marker Group Textual Conventions
--

PrtMarkerMarkTechTC ::= TEXTUAL-CONVENTION
-- This value is a type 2 enumeration
STATUS current
DESCRIPTION
"The type of marking technology used for this marking sub-unit"
SYNTAX INTEGER {
 other(1),
 unknown(2),
 electrophotographicLED(3),
 electrophotographicLaser(4),
 electrophotographicOther(5),
 impactMovingHeadDotMatrix9pin(6),
 impactMovingHeadDotMatrix24pin(7),
 impactMovingHeadDotMatrixOther(8),
 impactMovingHeadFullyFormed(9),
 impactBand(10),
 impactOther(11),
 inkjetAqueous(12),
 inkjetSolid(13),
 inkjetOther(14),
 pen(15),
 thermalTransfer(16),
 thermalSensitive(17),
 thermalDiffusion(18),
 thermalOther(19),
 electroerosion(20),
 electrostatic(21),
 photographicMicrofiche(22),
 photographicImagesetter(23),
 photographicOther(24),
 ionDeposition(25),
 eBeam(26),
 typesetter(27)
 }
 }

PrtMarkerCounterUnitTC ::= TEXTUAL-CONVENTION
-- This value is a type 1 enumeration
STATUS current
DESCRIPTION
"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."

SYNTAX INTEGER {
 tenThousandthsOfInches(3), -- .0001
 micrometers(4),
 characters(5),
 lines(6),
 impressions(7),
 sheets(8),
 dotRow(9),

```
        hours(11),  
        feet(16),  
        meters(17)  
    }
```

PrtMarkerSuppliesTypeTC ::= TEXTUAL-CONVENTION

```
-- This value is a type 3 enumeration  
STATUS      current  
DESCRIPTION  
    "The type of this supply."
```

```
SYNTAX      INTEGER {  
    other(1),  
    unknown(2),  
    toner(3),  
    wasteToner(4),  
    ink(5),  
    inkCartridge(6),  
    inkRibbon(7),  
    wasteInk(8),  
    opc(9), -- photo conductor  
    developer(10),  
    fuserOil(11),  
    solidWax(12),  
    ribbonWax(13),  
    wasteWax(14),  
    fuser(15),  
    coronaWire(16),  
    fuserOilWick(17),  
    cleanerUnit(18),  
    fuserCleaningPad(19),  
    transferUnit(20),  
    tonerCartridge(21),  
    fuserOiler(22)  
}
```

PrtMarkerSuppliesSupplyUnitTC ::= TEXTUAL-CONVENTION

```
-- This value is a type 1 enumeration  
STATUS      current  
DESCRIPTION  
    "Unit of this marker supply container/receptacle."
```

```
SYNTAX      INTEGER {  
    tenThousandthsOfInches(3), -- .0001  
    micrometers(4),  
    impressions(7),  
    sheets(8),  
    thousandthsOfOunces(12),  
    hours(11),  
    tenthsOfGrams(13),  
    hundrethsOfFluidOunces(14),  
    tenthsOfMilliliters(15),  
    feet(16),  
    meters(17)  
}
```

PrtMarkerSuppliesClassTC ::= TEXTUAL-CONVENTION

```
-- This value is a type 1 enumeration  
STATUS      current  
DESCRIPTION  
    "Indicates whether this supply entity represents a supply
```

```
        that is consumed or a receptacle that is filled."
SYNTAX      INTEGER {
                other(1),
                supplyThatIsConsumed(3),
                receptacleThatIsFilled(4)
            }

PrtMarkerColorantRoleTC ::= TEXTUAL-CONVENTION
-- This value is a type 1 enumeration
STATUS      current
DESCRIPTION
    "The role played by this colorant."
SYNTAX      INTEGER { -- Colorant Role
                other(1),
                process(3),
                spot(4)
            }

--
-- Media Path Textual Conventions
--

PrtMediaPathMaxSpeedPrintUnitTC OBJECT-TYPE
-- This value is a type 1 enumeration
STATUS      current
DESCRIPTION
    "The unit of measure used in specifying the speed of all
    media paths in the printer."
SYNTAX      INTEGER {
                tenThousandthsOfInchesPerHour(3), -- .0001/hour
                micrometersPerHour(4),
                charactersPerHour(5),
                linesPerHour(6),
                impressionsPerHour(7),
                sheetsPerHour(8),
                dotRowPerHour(9),
                feetPerHour(16),
                metersPerHour(17)
            }

PrtMediaPathTypeTC ::= TEXTUAL-CONVENTION
-- This value is a type 2 enumeration
STATUS      current
DESCRIPTION
    "The type of the media path for this media path."
SYNTAX      INTEGER {
                other(1),
                unknown(2),
                longEdgeBindingDuplex(3),
                shortEdgeBindingDuplex(4),
                simplex(5)
            }

--
-- Interpreter Group Textual Conventions
--

PrtInterpreterDefaultOrientationTC ::= TEXTUAL-CONVENTION
-- This value is a type 1 enumeration
STATUS      current
```

DESCRIPTION

"The current orientation default for this interpreter. This value may be overridden for a particular job (e.g., by a command in the input data stream)."

```
SYNTAX      INTEGER {
                other(1),
                portrait(3),
                landscape(4)
            }
```

PrtInterpreterTwoWayTC ::= TEXTUAL-CONVENTION

-- This is a type 1 enumeration

STATUS current

DESCRIPTION

"Indicates whether or not this interpreter returns information back to the host."

```
SYNTAX      INTEGER {
                yes(3),
                no(4)
            }
```

--
-- Console Group Textual Conventions
--

PrtConsoleColorTC ::= TEXTUAL-CONVENTION

-- This value is a type 2 enumeration

STATUS current

DESCRIPTION

"The color of this light."

```
SYNTAX      INTEGER {
                other(1),
                unknown(2),
                white(3),
                red(4),
                green(5),
                blue(6),
                cyan(7),
                magenta(8),
                yellow(9)
            }
```

--
-- Alert Group Textual Conventions
--

PrtAlertSeverityLevelTC ::= TEXTUAL-CONVENTION

-- This value is a type 1 enumeration

STATUS current

DESCRIPTION

"The level of severity of this alert table entry. The printer determines the severity level assigned to each entry into the table. A warning is defined to be a non-critical alert. The value also indicates whether the event is a binary change event or a unary change event."

```
SYNTAX      INTEGER {
                other(1),
                criticalBinaryChangeEvent(3),
            }
```

```
        warningUnaryChangeEvent(4),  
        warningBinaryChangeEvent(5)  
    }
```

PrtAlertTrainingLevelTC ::= TEXTUAL-CONVENTION

-- This value is a type 2 enumeration

STATUS current

DESCRIPTION

"The level of training required to handle this alert, if human intervention is required. The noInterventionRequired value should be used if the event does not require any human intervention. The training level is an enumeration that is determined and assigned by the printer manufacturer based on the information or the training required to handle this alert. The printer will break alerts into these different training levels. It is the responsibility of the management application in the system to determine how a particular alert is handled and how and to whom that alert is routed. The following are the four training levels of alerts:

Field Service - Alerts that typically require advanced training and technical knowledge of the printer and its sub-units. An example of a technical person would be a manufacturer's Field Service representative, or other person formally trained by the manufacturer or similar representative.

Trained - Alerts that require an intermediate or moderate level of knowledge of the printer and its sub-units. A typical examples of alerts that a trained operator can handle is replacing toner cartridges.

Untrained - Alerts that can be fixed without prior training either because the action to correct the alert is obvious or the printer can help the untrained person fix the problem. A typical example of such an alert is reloading paper trays and emptying output bins on a low end printer.

Management - Alerts that have to do with overall operation of and configuration of the printer. Examples of management events are configuration change of sub-units."

```
SYNTAX    INTEGER {  
            other(1),  
            unknown(2),  
            untrained(3),  
            trained(4),  
            fieldService(5),  
            management(6),  
            noInterventionRequired(7)  
        }
```

PrtAlertGroupTC ::= TEXTUAL-CONVENTION

-- This value is a type 1 enumeration for values in the range 1 to 29.

-- Values of 30 and greater are type 2 enumerations and are for use in

-- other MIBs that augment tables in the Printer MIB. Therefore, other

-- MIBs may assign alert codes of 30 or higher to use the alert table

-- from the Printer MIB without requiring revising and re-publishing this

```
-- document.  
STATUS      current  
DESCRIPTION  
    "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.  
    Wherever possible, these enumerations match the  
    sub-identifier that identifies the relevant table in the  
    printmib.  
  
    NOTE: Alert type codes have been added for the host resources MIB  
    storage table and device table. These additional types are for  
    for situations in which the printer's storage and device objects  
    must generate alerts (and possibly traps for critical alerts)."
```

```
SYNTAX      INTEGER {  
    other(1),  
    hostResourcesMIBStorageTable(3),  
    hostResourcesMIBDeviceTable(4),  
    generalPrinter(5),  
    cover(6),  
    localization(7),  
    input(8),  
    output(9),  
    marker(10),  
    markerSupplies(11),  
    markerColorant(12),  
    mediaPath(13),  
    channel(14),  
    interpreter(15),  
    consoleDisplayBuffer(16),  
    consoleLights(17),  
    alert(18)  
}
```

```
PrtAlertCodeTC ::= TEXTUAL-CONVENTION  
-- This value is a type 2 enumeration  
STATUS      current  
DESCRIPTION  
    "The code that describes the type of alert for this entry in  
    the table. Binary change event alerts describe states of the subunit,  
    while unary change event alerts describe a single event. The same  
    alert code can be used for a binary change event or a unary change  
    event, depending on implementation. Also, the same alert code can be  
    used to indicate a critical or a non-critical (warning) alert,  
    depending on implementation. The value of prtAlertSeverityLevel  
    specifies binary vs. unary and critical vs. non-critical for each  
    event for the implementation.
```

While there are some specific codes for many subunits, the generic codes should be used for most subunit alerts. The network management station can then query the subunit specified by prtAlertGroup to determine further subunit status and other subunit information.

An agent shall not add two entries to the alert table for the same event, one containing a generic event code and the other containing a specific event code; the agent shall add only one entry in the alert table for each event; either generic (preferred) or specific, not both.

Implementation of the unary change event
 alertRemovalOfBinaryChangeEvent(1801) is optional. When implemented,
 this alert code shall indicate to network management stations that
 the trailing edge of a binary change event has occurred and the
 corresponding alert entry has been removed from the alert table. As
 with all events, the alertRemovalOfBinaryChangeEvent(1801) alert
 shall be placed at the end of the alert table. Such an alert table
 entry shall specify the following information:

```

prAlertSeverityLevel    warningUnaryChangeEvent(4)
prAlertTrainingLevel    noInterventionRequired(7)
prAlertGroup            alert(18)
prAlertGroupIndex      the index of the row in the alert
                        table of the binary change event
                        that this event has removed.
prAlertLocation         (-2)
prAlertCode             alertRemovalOfBinaryChangeEvent(1801)
prAlertDescription      <description or null string>
prAlertTime             (Optional) the value of sysUpTime at
                        the time of the removal of the binary
                        change event from the alert table.
  
```

Optionally, the agent may generate a trap coincident with removing
 the binary change event and placing the unary change event
 alertRemovalOfBinaryChangeEvent(1801) in the alert table.
 For such a trap, the prAlertIndex sent with the above trap
 parameters shall be the index of the alertRemovalOfBinaryChangeEvent
 row that was added to the prAlertTable; not the index of the row
 that was removed from the prAlertTable."

```

SYNTAX    INTEGER {
    other(1),          -- an event that is not represented by one of
                      -- the alert codes specified below.
    unknown(2),
      -- The following generic codes are common to multiple
      -- groups. The NMS may examine the prAlertGroup
      -- object to determine what group to query for further
      -- information.
    coverOpened(3),
    coverClosed(4),
    interlockOpened(5),
    interlockClosed(6),
    configurationChanged(7),
    jammed(8),
    subunitMissing(9),          -- The subunit (tray,bin,etc.)
                                -- has been removed.
    subunitLifeAlmostOver(10),
    subunitLifeOver(11),
    subunitAlmostEmpty(12),
    subunitEmpty(13),
    subunitAlmostFull(14),
    subunitFull(15),
    subunitNearLimit(16),
    subunitAtLimit(17),
    subunitOpened(18),
    subunitClosed(19),
    subunitTurnedOn(20),
    subunitTurnedOff(21),
    subunitOffline(22),
    subunitPowerSaver(23),
    subunitWarmingUp(24),
  
```

```
subunitAdded(25),
subunitRemoved(26),
subunitResourceAdded(27),
subunitResourceRemoved(28),
subunitRecoverableFailure(29),
subunitUnrecoverableFailure(30),
subunitRecoverableStorageError(31),
subunitUnrecoverableStorageError(32),
subunitMotorFailure(33),
subunitMemoryExhausted(34),
    -- general Printer group
    coverOpened(501),
    coverClosed(502),
    poweredUp(503),
    poweredDown(504),
printerNMSReset(505), -- The printer has been reset by
    -- some network management station
    -- (NMS) writing into
    -- 'prtGeneralReset'. The value
    -- written shall be stored as the
    -- value of the prtAlertLocation
    -- object indicating the type of
    -- reset: powerCycleReset(4),
    -- resetToNVRAM(5),
    -- resetToFactoryDefaults(6), etc.
printerManualReset(506),
    -- The printer has been reset manually. The
    -- value of prtAlertLocation may be used to
    -- indicate the type of reset.
printerReadyToPrint(507),
    -- The printer is ready to print. (i.e., not
    -- warming up, not in power save state, not
    -- adjusting print quality, etc.).
    -- Input Group
    inputMediaTrayMissing(801),
    inputMediaSizeChanged(802),
    inputMediaWeightChanged(803),
    inputMediaTypeChanged(804),
    inputMediaColorChanged(805),
    inputMediaFormPartsChange(806),
    inputMediaSupplyLow(807),
    inputMediaSupplyEmpty(808),
inputMediaChangeRequest(809),
    -- An interpreter has detected that a
    -- different medium is need in this input
    -- tray subunit. The prtAlertDescription may
    -- be used to convey a human readable
    -- description of the medium required to
    -- satisfy the request.
inputManualInputRequest(810),
    -- An interpreter has detected that manual
    -- input is required in this subunit. The
    -- prtAlertDescription may be used to convey
    -- a human readable description of the medium
    -- required to satisfy the request.
inputTrayPositionFailure(811),
    -- The input tray failed to position
    -- correctly.
inputTrayElevationFailure(812),
inputCannotFeedSizeSelected(813),
```

```
    -- Output Group
    outputMediaTrayMissing(901),
    outputMediaTrayAlmostFull(902),
    outputMediaTrayFull(903),
    outputMailboxSelectFailure(904),
    -- Marker group
    markerFuserUnderTemperature(1001),
    markerFuserOverTemperature(1002),
    markerFuserTimingFailure(1003),
    markerFuserThermistorFailure(1004),
    markerAdjustingPrintQuality(1005),
    -- Marker Supplies group
    markerTonerEmpty(1101),
    markerInkEmpty(1102),
    markerPrintRibbonEmpty(1103),
    markerTonerAlmostEmpty(1104),
    markerInkAlmostEmpty(1105),
    markerPrintRibbonAlmostEmpty(1106),
    markerWasteTonerReceptacleAlmostFull(1107),
    markerWasteInkReceptacleAlmostFull(1108),
    markerWasteTonerReceptacleFull(1109),
    markerWasteInkReceptacleFull(1110),
    markerOpcLifeAlmostOver(1111),
    markerOpcLifeOver(1112),
    markerDeveloperAlmostEmpty(1113),
    markerDeveloperEmpty(1114),
    markerTonerCartridgeMissing(1115),
    -- Media Path Device Group
    mediaPathMediaTrayMissing(1301),
    mediaPathMediaTrayAlmostFull(1302),
    mediaPathMediaTrayFull(1303),
    -- interpreter Group
    interpreterMemoryIncreased(1501),
    interpreterMemoryDecreased(1502),
    interpreterCartridgeAdded(1503),
    interpreterCartridgeDeleted(1504),
    interpreterResourceAdded(1505),
    interpreterResourceDeleted(1506),
    interpreterResourceUnavailable(1507),
    interpreterComplexPageEncountered(1509),
    -- The interpreter has encountered a page that
    -- is too complex for the resources that are
    -- available.
    --
    -- Alert Group
    --
    alertRemovalOfBinaryChangeEntry(1801)
    -- A binary change event entry has been removed
    -- from the alert table. This unary change alert
    -- table entry is added to the end of the alert
    -- table.
}
```

```
-- The 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.
```

```
--
-- Implementation of every object in this group is mandatory.

prtGeneral OBJECT IDENTIFIER ::= { printmib 5 }

prtGeneralTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PrtGeneralEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table of general information per printer.
        Objects in this table are defined in various
        places in the MIB, nearby the groups to
        which they apply.  They are all defined
        here to minimize the number of tables that would
        otherwise need to exist."
    ::= { prtGeneral 1 }

prtGeneralEntry OBJECT-TYPE
    SYNTAX      PrtGeneralEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry exists in this table for each
        device entry in the hostmib device table with a device type
        of 'printer'"
    INDEX      { hrDeviceIndex }
    ::= { prtGeneralTable 1 }

PrtGeneralEntry ::= SEQUENCE {
    -- Note that not all of the objects in this sequence are in the
    -- general printer group.
    prtGeneralConfigChanges      Counter32,
    prtGeneralCurrentLocalization Integer32,
    prtGeneralReset              PrtGeneralResetTC,
    prtGeneralCurrentOperator    OCTET STRING,
    prtGeneralServicePerson      OCTET STRING,
    prtInputDefaultIndex         Integer32,
    prtOutputDefaultIndex        Integer32,
    prtMarkerDefaultIndex        Integer32,
    prtMediaPathDefaultIndex     Integer32,
    prtConsoleLocalization       Integer32,
    prtConsoleNumberOfDisplayLines Integer32,
    prtConsoleNumberOfDisplayChars Integer32,
    prtConsoleDisable            INTEGER,
    prtGeneralStartupPage        PresentOnOff,
    prtGeneralBannerPage         PresentOnOff,
    prtGeneralAdminName          OCTET STRING,
    prtGeneralSerialNumber       OCTET STRING,
    prtAlertCriticalEvents       Counter32,
    prtAlertAllEvents            Counter32
}

prtGeneralConfigChanges OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Counts configuration changes that change the capabilities of
        a printer, such as the addition/deletion of input/output bins,
```

the addition/deletion of interpreters, or changes in media size. Such changes will often affect the capability of the printer to service certain types of print jobs.

Management applications may cache infrequently changed configuration information about sub-units on the printer. This object should be incremented whenever the agent wishes such applications to invalidate that cache and re-download all of this configuration information, thereby signaling a change in the printer's configuration.

For example, if an input tray that contained paper of different dimensions was added, this counter would be incremented.

As an additional example, this counter would not be incremented when an input tray is removed or the level of an input device changes."

```
::= { prtGeneralEntry 1 }
```

prtGeneralCurrentLocalization OBJECT-TYPE

```
SYNTAX      Integer32 (1..65535)
MAX-ACCESS  read-write
MIN-ACCESS  read-only
STATUS      current
```

DESCRIPTION

"The value of the prtLocalizationIndex corresponding to the current language, country, and character set to be used for localized string values that are identified as being dependent on the value of this object. Note that this object does not apply to localized strings in the prtConsole group or to any object that is not explicitly identified as being localized according to prtGeneralCurrentLocalization."

```
::= { prtGeneralEntry 2 }
```

prtGeneralReset OBJECT-TYPE

```
-- This value is a type 3 enumeration
```

```
SYNTAX      PrtGeneralResetTC
MAX-ACCESS  read-write
STATUS      current
```

DESCRIPTION

"Setting this value to 'powerCycleReset', 'resetToNVRAM', or 'resetToFactoryDefaults' will result in the resetting of the printer. When read, this object will always have the value 'notResetting(3)', and a SET of the value 'notResetting' shall have no effect on the printer. Some of the defined values are optional. However, every implementation must support at least the values 'notResetting' and 'resetToNVRAM'."

```
::= { prtGeneralEntry 3 }
```

```
-- The Cover Table
```

```
--
```

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

```
prtCover      OBJECT IDENTIFIER ::= { printmib 6 }
```

prtCoverTable OBJECT-TYPE

```
SYNTAX      SEQUENCE OF PrtCoverEntry
```

```
MAX-ACCESS not-accessible
STATUS      current
DESCRIPTION
    "A table of the covers and interlocks of the printer."
 ::= { prtCover 1 }

prtCoverEntry OBJECT-TYPE
SYNTAX      PrtCoverEntry
MAX-ACCESS not-accessible
STATUS      current
DESCRIPTION
    "Information about a cover or interlock.
     Entries may exist in the table for each device
     index with a device type of 'printer'."
INDEX       { hrDeviceIndex, prtCoverIndex }
 ::= { prtCoverTable 1 }

PrtCoverEntry ::= SEQUENCE {
    prtCoverIndex      Integer32,
    prtCoverDescription OCTET STRING,
    prtCoverStatus     INTEGER
}

prtCoverIndex OBJECT-TYPE
SYNTAX      Integer32 (1..65535)
MAX-ACCESS not-accessible
STATUS      current
DESCRIPTION
    "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."
 ::= { prtCoverEntry 1 }

prtCoverDescription OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE(0..255))
MAX-ACCESS read-only
STATUS      current
DESCRIPTION
    "The manufacturer provided cover sub-mechanism name in the
     localization specified by prtGeneralCurrentLocalization."
 ::= { prtCoverEntry 2 }

prtCoverStatus OBJECT-TYPE
SYNTAX      PrtCoverStatusTC
-- This value is a type 2 enumeration
MAX-ACCESS read-only
STATUS      current
DESCRIPTION
    "The status of this cover sub-unit."
 ::= { prtCoverEntry 3 }

-- The Localization Table
--
-- 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.

prtLocalization OBJECT IDENTIFIER ::= { printmib 7 }

prtLocalizationTable OBJECT-TYPE
SYNTAX SEQUENCE OF PrtLocalizationEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The available localizations in this printer."
::= { prtLocalization 1 }

prtLocalizationEntry OBJECT-TYPE
SYNTAX PrtLocalizationEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"A description of a localization.
Entries may exist in the table for each device
index with a device type of 'printer'."
INDEX { hrDeviceIndex, prtLocalizationIndex }
::= { prtLocalizationTable 1 }

PrtLocalizationEntry ::= SEQUENCE {
prtLocalizationIndex Integer32,
prtLocalizationLanguage OCTET STRING,
prtLocalizationCountry OCTET STRING,
prtLocalizationCharacterSet CodedCharSet
}

prtLocalizationIndex OBJECT-TYPE
SYNTAX Integer32 (1..65535)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"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."
::= { prtLocalizationEntry 1 }

prtLocalizationLanguage OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(0..2))
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"A two character language code from ISO 639. Examples EN, GB,
CA, FR, DE."
::= { prtLocalizationEntry 2 }

prtLocalizationCountry OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(0..2))
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"A two character country code from ISO 3166, a blank string
(two space characters) shall indicate that the country is
not defined. Examples: US, FR, DE, ..."

```
 ::= { prtLocalizationEntry 3 }

prtLocalizationCharacterSet OBJECT-TYPE
    SYNTAX      CodedCharSet
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The coded character set used for this localization."
    ::= { prtLocalizationEntry 4 }

-- The System Resources Tables

-- The Printer MIB makes use of the Host MIB to
-- define system resources by referencing the storage
-- and device groups of the print group.  In order to
-- determine, amongst multiple printers serviced by
-- one agent, which printer owns a particular
-- resource, the prtStorageRef and prtDeviceRef tables
-- associate particular storage and device entries to
-- printers.

prtStorageRefTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PrtStorageRefEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        ""
    ::= { prtGeneral 2 }

prtStorageRefEntry OBJECT-TYPE
    SYNTAX      PrtStorageRefEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table will have an entry for each entry in
        the host MIB storage table that represents storage associated
        with a printer managed by this agent."
    INDEX      { hrStorageIndex, prtStorageRefSeqNumber }
    ::= { prtStorageRefTable 1 }

PrtStorageRefEntry ::= SEQUENCE {
    prtStorageRefSeqNumber  Integer32,
    prtStorageRefIndex      Integer32
}

prtStorageRefSeqNumber OBJECT-TYPE
    SYNTAX      Integer32 (0..65535)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This value will be unique amongst all entries with a common
        value of hrStorageIndex.

        This object allows a storage entry to point to the multiple
        printer devices with which it is associated."

    ::= { prtStorageRefEntry 1 }

prtStorageRefIndex OBJECT-TYPE
    SYNTAX      Integer32 (0..65535)
```

```
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The value of the hrDeviceIndex of the printer device that this
    storageEntry is associated with."
 ::= { prtStorageRefEntry 2 }

prtDeviceRefTable OBJECT-TYPE
SYNTAX SEQUENCE OF PrtDeviceRefEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    ""
 ::= { prtGeneral 3 }

prtDeviceRefEntry OBJECT-TYPE
SYNTAX PrtDeviceRefEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "This table will have an entry for each entry in
    the host MIB device table that represents a device associated
    with a printer managed by this agent."
INDEX { hrDeviceIndex, prtDeviceRefSeqNumber }
 ::= { prtDeviceRefTable 1 }

PrtDeviceRefEntry ::= SEQUENCE {
    prtDeviceRefSeqNumber Integer32,
    prtDeviceRefIndex Integer32
}

prtDeviceRefSeqNumber OBJECT-TYPE
SYNTAX Integer32 (0..65535)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "This value will be unique amongst all entries with a common
    value of hrDeviceIndex.

    This object allows a device entry to point to the multiple
    printer devices with which it is associated."
 ::= { prtDeviceRefEntry 1 }

prtDeviceRefIndex OBJECT-TYPE
SYNTAX Integer32 (0..65535)
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The value of the hrDeviceIndex of the printer device that this
    deviceEntry is associated with."
 ::= { prtDeviceRefEntry 2 }

-- The Responsible Party group
--
-- This group is optional. However, to claim conformance to this
-- group, it is necessary to implement every object in the group.

prtGeneralCurrentOperator OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(0..127))
MAX-ACCESS read-write
```

MIN-ACCESS read-only
STATUS current
DESCRIPTION

"The name of the current human operator responsible for operating this printer. It is suggested that this string include information that would enable other humans to reach the operator, such as a phone number. As a convention to facilitate automatic notification of the operator by the agent or the network management station, the phone number, fax number or email address should be placed on a separate line starting with ASCII LF (hex 0C) and the ASCII text string (without the quotes): 'phone: ', 'fax: ', and 'email: ', respectively. Phone numbers may contain digits, whitespace and parentheses, which shall be ignored. Phone numbers may also include ASCII comma characters(hex 2C) that are used to indicate a half-second pause during the dialing sequence.

NOTE: For interoperability purposes, it is advisable to use email addresses formatted according to RFC 822 requirements."

::= { prtGeneralEntry 4 }

prtGeneralServicePerson OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(0..127))
MAX-ACCESS read-write
MIN-ACCESS read-only
STATUS current
DESCRIPTION

"The name of the last human responsible for servicing this printer. It is suggested that this string include information that would enable other humans to reach the service person, such as a phone number. As a convention to facilitate automatic notification of the service person by the agent or a network management station, the phone number, fax number or email address should be placed on a separate line starting with ASCII LF (hex 0C) and the ASCII text string (without the quotes): 'phone: ', 'fax: ', and 'email: ', respectively. Phone numbers may contain digits, whitespace and parentheses, which shall be ignored. Phone numbers can also include one or more ASCII comma characters (hex 2C) to indicate a half-second pause during the dialing sequence.

NOTE: For interoperability purposes, it is advisable to use email addresses formatted according to RFC 822 requirements."

::= { prtGeneralEntry 5 }

-- The Auxiliary Sheet Group
--
-- The auxiliary sheet group allows the administrator to control the
-- production of auxiliary sheets by the printer.
--
-- This group is optional. However, to claim conformance to this group,
-- it is necessary to implement every object in the group.

prtGeneralStartupPage OBJECT-TYPE
SYNTAX PresentOnOff
MAX-ACCESS read-write

```
MIN-ACCESS read-only
STATUS current
DESCRIPTION
    "Used to enable or disable printing a startup page. If
    enabled, a startup page will be printed shortly after
    power-up, when the device is ready. Typical startup
    pages include test patterns and/or printer configuration
    information."
::= { prtGeneralEntry 16 }

prtGeneralBannerPage OBJECT-TYPE
SYNTAX PresentOnOff
MAX-ACCESS read-write
MIN-ACCESS read-only
STATUS current
DESCRIPTION
    "Used to enable or disable printing banner pages at the
    beginning of jobs. This is a master switch which applies
    to all jobs, regardless of interpreter."
::= { prtGeneralEntry 17 }

prtGeneralAdminName OBJECT-TYPE
SYNTAX PresentOnOff
MAX-ACCESS read-write
MIN-ACCESS read-only
STATUS current
DESCRIPTION
    "An administrator-specified name for an individual that maintains
    authoratative control over this device with regards to device
    operation and/or configuration."
::= { prtGeneralEntry 18 }

prtGeneralSerialNumber OBJECT-TYPE
SYNTAX PresentOnOff
MAX-ACCESS read-write
MIN-ACCESS read-only
STATUS current
DESCRIPTION
    "A recorded serial number for this device that indexes some type
    device catalog or inventory. This value is usually set by the
    device manufacturer but the MIB supports the option of
    writeability for this object for site-specific administration of
    device inventory or tracking."
::= { prtGeneralEntry 19 }

prtAlertCriticalEvents OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "A running counter of the number of critical alert events that
    have been recorded in the alert table"
::= { prtGeneralEntry 20 }

prtAlertAllEvents OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
```

```
STATUS      current
DESCRIPTION
"A running counter of the total number of alert event entries
(critical and non-critical) that have been recorded in the
alert table"
::= { prtGeneralEntry 21 }

-- The Input Group
--
-- 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.
--
-- Implementation of every object in this group is mandatory.

prtInput    OBJECT IDENTIFIER ::= { printmib 8 }

prtInputDefaultIndex OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-write
MIN-ACCESS  read-only
STATUS      current
DESCRIPTION
"The value of prtInputIndex corresponding to the default input
sub-unit: that is, this object selects the default source of
input media.

This value shall be -1 if there is no default input subunit specified
for the printer as a whole.  In this case, the actual default
input subunit may be specified by means outside the scope of this
MIB, such as by each interpreter in a printer with multiple
interpreters."

::= { prtGeneralEntry 6 }

prtInputTable OBJECT-TYPE
SYNTAX      SEQUENCE OF PrtInputEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"A table of the devices capable of providing media for input
to the printing process."
::= { prtInput 2 }

prtInputEntry OBJECT-TYPE
SYNTAX      PrtInputEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"Attributes of a device capable of providing media for input
to the printing process.

Entries may exist in the table for each device
index with a device type of 'printer'."
```

```
INDEX { hrDeviceIndex, prtInputIndex }  
 ::= { prtInputTable 1 }
```

```
PrtInputEntry ::= SEQUENCE {  
    prtInputIndex          Integer32,  
    prtInputType          PrtInputTypeTC,  
    prtInputDimUnit       PrtMediaUnitTC,  
    prtInputMediaDimFeedDirDeclared Integer32,  
    prtInputMediaDimXFeedDirDeclared Integer32,  
    prtInputMediaDimFeedDirChosen   Integer32,  
    prtInputMediaDimXFeedDirChosen   Integer32,  
    prtInputCapacityUnit   PrtCapacityUnitTC,  
    prtInputMaxCapacity     Integer32,  
    prtInputCurrentLevel    Integer32,  
    prtInputStatus          PrtSubUnitStatusTC,  
    prtInputMediaName       OCTET STRING,  
    prtInputName            OCTET STRING,  
    prtInputVendorName      OCTET STRING,  
    prtInputModel           OCTET STRING,  
    prtInputVersion         OCTET STRING,  
    prtInputSerialNumber    OCTET STRING,  
    prtInputDescription     OCTET STRING,  
    prtInputSecurity        PresentOnOff,  
    prtInputMediaWeight     Integer32,  
    prtInputMediaType       OCTET STRING,  
    prtInputMediaColor      OCTET STRING,  
    prtInputMediaFormParts  Integer32,  
    prtInputManualFeedTimeout Integer32,  
    prtInputAutoSwitch      PresentOnOff,  
    prtInputNextIndex       Integer32  
}
```

```
prtInputIndex OBJECT-TYPE  
    SYNTAX      Integer32 (1..65535)  
    MAX-ACCESS not-accessible  
    STATUS      current  
    DESCRIPTION  
        "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."  
    ::= { prtInputEntry 1 }
```

```
prtInputType OBJECT-TYPE  
    SYNTAX PrtInputTypeTC  
    MAX-ACCESS read-only  
    STATUS      current  
    DESCRIPTION  
        "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."  
    ::= { prtInputEntry 2 }
```

```
prtInputDimUnit OBJECT-TYPE  
    SYNTAX      PrtMediaUnitTC  
    MAX-ACCESS read-only  
    STATUS      current  
    DESCRIPTION
```

"The unit of measurement for use calculating and relaying dimensional values for this input sub-unit."
 ::= { prtInputEntry 3 }

prtInputMediaDimFeedDirDeclared OBJECT-TYPE

SYNTAX Integer32
MAX-ACCESS read-write
MIN-ACCESS read-only
STATUS current
DESCRIPTION

"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 fed on this sub-unit. This dimension is measured in input sub-unit dimensional units (prtInputDimUnit). If this input sub-unit can reliably sense this value, the value is sensed by the printer and may not be changed by management requests. Otherwise, the value may be changed. The value (-1) means other and specifically means that this sub-unit places no restriction on this parameter. The value (-2) indicates unknown."

::= { prtInputEntry 4 }

prtInputMediaDimXFeedDirDeclared OBJECT-TYPE

SYNTAX Integer32
MAX-ACCESS read-write
STATUS current
DESCRIPTION

"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 (prtInputDimUnit). If this input sub-unit can reliably sense this value, the value is sensed by the printer and may not be changed by management requests. Otherwise, the value may be changed. The value (-1) means other and specifically means that this sub-unit places no restriction on this parameter. The value (-2) indicates unknown."

::= { prtInputEntry 5 }

prtInputMediaDimFeedDirChosen OBJECT-TYPE

SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"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 relative to the feed direction associated with that sub-unit and are in input sub-unit dimensional units (MediaUnit). 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."

```
::= { prtInputEntry 6 }
```

prtInputMediaDimXFeedDirChosen OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"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 (MediaUnit). 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."

```
::= { prtInputEntry 7 }
```

prtInputCapacityUnit OBJECT-TYPE

SYNTAX PprtCapacityUnitTC

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The unit of measurement for use in calculating and relaying capacity values for this input sub-unit."

```
::= { prtInputEntry 8 }
```

prtInputMaxCapacity OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-write

MIN-ACCESS read-only

STATUS current

DESCRIPTION

"The maximum capacity of the input sub-unit in input sub-unit capacity units (PprtCapacityUnitTC). 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 may not be changed by management requests; otherwise, the value may be written (by a Remote Control 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."

```
::= { prtInputEntry 9 }
```

prtInputCurrentLevel OBJECT-TYPE

SYNTAX Integer32 -- in capacity units (PprtCapacityUnitTC).

MAX-ACCESS read-write

MIN-ACCESS read-only

STATUS current

DESCRIPTION

"The current capacity of the input sub-unit in input sub-unit capacity units (PprtCapacityUnitTC). If this input sub-unit can reliably sense this value, the value is


```
 ::= { prtInputEntry 13 }

prtInputVendorName OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE(0..63))
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The vendor name of this input sub-unit."
 ::= { prtInputEntry 14 }

prtInputModel OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE(0..63))
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The model name of this input sub-unit."
 ::= { prtInputEntry 15 }

prtInputVersion OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE(0..63))
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The version of this input sub-unit."
 ::= { prtInputEntry 16 }

prtInputSerialNumber OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE(0..32))
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The serial number assigned to this input sub-unit."
 ::= { prtInputEntry 17 }

prtInputDescription OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE(0..255))
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "A free-form text description of this input
    sub-unit in the localization specified by
    prtGeneralCurrentLocalization."
 ::= { prtInputEntry 18 }

prtInputSecurity OBJECT-TYPE
SYNTAX      PresentOnOff
MAX-ACCESS  read-write
MIN-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Indicates if this input sub-unit has some security
    associated with it."
 ::= { prtInputEntry 19 }

-- The Input Media Group
--
-- The Input Media 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 Input Media group consists of a set of optional
-- "columns" in the Input Table. In this manner, a minimally
-- conforming implementation may choose to not support reporting
-- of media resources if it cannot do so.
--
-- This group is optional. However, to claim conformance to this
-- group, it is necessary to implement every object in the group.

```
prtInputMediaWeight OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-write
    MIN-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The weight of the medium associated with this input
        sub-unit in grams / per meter squared. The value (-2) means
        unknown."
    ::= { prtInputEntry 20 }
```

```
prtInputMediaType OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(0..63))
    MAX-ACCESS  read-write
    MIN-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The name of the type of medium associated with this input
        sub-unit. This name need not be processed by the printer; it
        might simply be displayed to an operator. The standardized
        string values from ISO 10175 (DPA) and ISO 10180 (SPDL) are:

        stationery      Separately cut sheets of an opaque material
        transparency    Separately cut sheets of a transparent material
        envelope        Envelopes that can be used for conventional
        mailing purposes
        envelope-plain  Envelopes that are not preprinted and have no
        windows
        envelope-window Envelopes that have windows for addressing
        purposes
        continuous-long  Continuously connected sheets of an opaque
        material connected along the long edge
        continuous-short Continuously connected sheets of an opaque
        material connected along the short edge
        tab-stock       Media with tabs
        multi-part-form  Form medium composed of multiple layers not
        pre-attached to one another; each sheet may be
        drawn separately from an input source
        labels          Label stock
        multi-layer     Form medium composed of multiple layers which
        are pre-attached to one another; e.g., for
        use with impact printers.

        Implementors may add additional string values. The naming
        conventions in ISO 9070 are recommended in order to avoid
        potential name clashes."
```

```
::= { prtInputEntry 21 }
```

```
prtInputMediaColor OBJECT-TYPE
```

```
SYNTAX      OCTET STRING (SIZE(0..63))
MAX-ACCESS  read-write
MIN-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The name of the color of the medium associated with
    this input sub-unit using standardized string values
    from ISO 10175 (DPA) and ISO 10180 (SPDL) which are:

    other
    unknown
    white
    pink
    yellow
    buff
    goldenrod
    blue
    green
    transparent

    Implementors may add additional string values. The naming
    conventions in ISO 9070 are recommended in order to avoid
    potential name clashes."
 ::= { prtInputEntry 22 }
```

```
prtInputMediaFormParts OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-write
MIN-ACCESS  read-only
STATUS      current
DESCRIPTION
    "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
    specifically indicates that the device places no
    restrictions on this parameter. The value (-2) means
    unknown."
 ::= { prtInputEntry 23 }
```

```
-- The Input Switching Group
--
-- The input switching group allows the administrator to set the manual
-- input subunit timeout for the printer and to control the automatic
-- input subunit switching by the printer when an input subunit becomes
-- empty.
--
-- This group is optional. However, to claim conformance to this group,
-- it is required to implement every object in the group.
```

```
prtInputManualFeedTimeout OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-write
MIN-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The duration (in seconds) after which the printer shall
    either:

    (a) switch to another input subunit, if the value of
```

prtInputNextIndex is non-zero and prtInputAutoSwitch
is on(3)
or
(b) abort any job waiting for manually fed input, if the
value of prtInputNextIndex is zero or
prtInputAutoSwitch is off(4) or notPresent(5).

The event which causes the printer to enter the waiting state
is product specific. A value of (-1) implies 'other' or
'infinite' which translates to 'this input subunit doesn't
support manual feed'. A value of (-2) implies 'unknown'."

::= { prtInputEntry 24 }

prtInputAutoSwitch OBJECT-TYPE

SYNTAX PresentOnOff

MAX-ACCESS read-write

MIN-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates the state of the auto input switching feature. The
value notPresent(5) indicates the feature is not currently
supported. Exact behavior of this feature is product specific."

::= { prtInputEntry 25 }

prtInputNextIndex OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-write

MIN-ACCESS read-only

STATUS current

DESCRIPTION

"The value of prtInputIndex corresponding to the input subunit
which will be used when this input subunit is emptied. A value
of zero(0) indicates that auto input switching will not occur
when this input subunit is emptied."

::= { prtInputEntry 26 }

-- The Output Group

--

-- 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.

--

-- Implementation of every object in this group is mandatory.

prtOutput OBJECT IDENTIFIER ::= { printmib 9 }

prtOutputDefaultIndex OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-write

MIN-ACCESS read-only

STATUS current

DESCRIPTION

"The value of prtOutputIndex corresponding to the default
output sub-unit; that is, this object selects the default
output destination."

This value shall be -1 if there is no default output subunit specified for the printer as a whole. In this case, the actual default output subunit may be specified by means outside the scope of this MIB, such as by each interpreter in a printer with multiple interpreters."

```
::= { prtGeneralEntry 7 }
```

```
prtOutputTable OBJECT-TYPE
  SYNTAX      SEQUENCE OF PrtOutputEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "A table of the devices capable of receiving media delivered
    from the printing process."
  ::= { prtOutput 2 }
```

```
prtOutputEntry OBJECT-TYPE
  SYNTAX      PrtOutputEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "Attributes of a device capable of receiving media delivered
    from the printing process.

    Entries may exist in the table for each device
    index with a device type of 'printer'."
  INDEX { hrDeviceIndex, prtOutputIndex }
  ::= { prtOutputTable 1 }
```

```
PrtOutputEntry ::= SEQUENCE {
  prtOutputIndex          Integer32,
  prtOutputType          PrtOutputTypeTC,
  prtOutputCapacityUnit  PrtCapacityUnitTC,
  prtOutputMaxCapacity   Integer32,
  prtOutputRemainingCapacity Integer32,
  prtOutputStatus        PrtSubUnitStatusTC,
  prtOutputName          OCTET STRING,
  prtOutputVendorName    OCTET STRING,
  prtOutputModel         OCTET STRING,
  prtOutputVersion       OCTET STRING,
  prtOutputSerialNumber  OCTET STRING,
  prtOutputDescription   OCTET STRING,
  prtOutputSecurity       PresentOnOff,
  prtOutputDimUnit       PrtMediaUnitTC,
  prtOutputMaxDimFeedDir Integer32,
  prtOutputMaxDimXFeedDir Integer32,
  prtOutputMinDimFeedDir Integer32,
  prtOutputMinDimXFeedDir Integer32,
  prtOutputStackingOrder PrtOutputStackingOrderTC,
  prtOutputPageDeliveryOrientation PrtOutputPageDeliveryOrientationTC,
  prtOutputBursting      PresentOnOff,
  prtOutputDecollating   PresentOnOff,
  prtOutputPageCollated  PresentOnOff,
  prtOutputOffsetStacking PresentOnOff
}
```

```
prtOutputIndex OBJECT-TYPE
  SYNTAX      Integer32
```

```
MAX-ACCESS not-accessible
STATUS      current
DESCRIPTION
    "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 devices to the printer), values
    are expected to remain stable across successive printer
    power cycles."
 ::= { prtOutputEntry 1 }
```

```
prtOutputType OBJECT-TYPE
-- This value is a type 2 enumeration
SYNTAX      PrtOutputTypeTC
MAX-ACCESS read-only
STATUS      current
DESCRIPTION
    "The type of technology supported by this output sub-unit."
 ::= { prtOutputEntry 2 }
```

```
prtOutputCapacityUnit OBJECT-TYPE
SYNTAX      PrtCapacityUnitTC
MAX-ACCESS read-only
STATUS      current
DESCRIPTION
    "The unit of measurement for use in calculating and relaying
    capacity values for this output sub-unit."
 ::= { prtOutputEntry 3 }
```

```
prtOutputMaxCapacity OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS read-write
MIN-ACCESS read-only
STATUS      current
DESCRIPTION
    "The maximum capacity of this output sub-unit in output
    sub-unit capacity units (PrtCapacityUnitTC). 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 may not be changed by management
    requests; otherwise, the value may be written
    (by a Remote Control 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."
 ::= { prtOutputEntry 4 }
```

```
prtOutputRemainingCapacity OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS read-write
MIN-ACCESS read-only
STATUS      current
DESCRIPTION
    "The remaining capacity of the possible output sub-unit
    capacity in output sub-unit capacity units (PrtCapacityUnitTC)
    of this output sub-unit. If this output sub-unit can
    reliably sense this value, the value is sensed by the
    printer and may not be modified by management requests;
    otherwise, the value may be written (by
```



```
prtOutputVersion OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(0..63))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The version of this output sub-unit."
    ::= { prtOutputEntry 10 }

prtOutputSerialNumber OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(0..63))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The serial number assigned to this output sub-unit."
    ::= { prtOutputEntry 11 }

prtOutputDescription OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(0..255))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        " A free-form text description of this output sub-unit in the
        localization specified by prtGeneralCurrentLocalization."
    ::= { prtOutputEntry 12 }

prtOutputSecurity OBJECT-TYPE
    SYNTAX      PresentOnOff
    MAX-ACCESS  read-write
    MIN-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Indicates if this output sub-unit has some security associated
        with it and if that security is enabled or not."
    ::= { prtOutputEntry 13 }

-- The Output Dimensions Group
--
-- This group is optional.  However, to claim conformance to this
-- group, it is necessary to implement every object in the group.

prtOutputDimUnit OBJECT-TYPE
    SYNTAX      PrtMediaUnitTC
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The unit of measurement for use in calculating and relaying
        dimensional values for this output sub-unit."
    ::= { prtOutputEntry 14 }

prtOutputMaxDimFeedDir OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-write
    MIN-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "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 (MediaUnit).  If this output
```

sub-unit can reliably sense this value, the value is sensed by the printer and may not be changed with management protocol operations."

::= { prtOutputEntry 15 }

prtOutputMaxDimXFeedDir OBJECT-TYPE

SYNTAX Integer32
MAX-ACCESS read-write
MIN-ACCESS read-only
STATUS current

DESCRIPTION

"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 (MediaUnit). If this output sub-unit can reliably sense this value, the value is sensed by the printer and may not be changed with management protocol operations."

::= { prtOutputEntry 16 }

prtOutputMinDimFeedDir OBJECT-TYPE

SYNTAX Integer32
MAX-ACCESS read-write
MIN-ACCESS read-only
STATUS current

DESCRIPTION

"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 may not be changed with management protocol operations."

::= { prtOutputEntry 17 }

prtOutputMinDimXFeedDir OBJECT-TYPE

SYNTAX Integer32
MAX-ACCESS read-write
MIN-ACCESS read-only
STATUS current

DESCRIPTION

"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 may not be changed with management protocol operations."

::= { prtOutputEntry 18 }

-- The Output Features Group

--

-- This group is optional. However, to claim conformance to this group, it is necessary to implement every object in the group.

prtOutputStackingOrder OBJECT-TYPE

-- This value is a type 1 enumeration
SYNTAX PrtOutputStackingOrderTC
MAX-ACCESS read-write
MIN-ACCESS read-only

```
STATUS      current
DESCRIPTION
    "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."
 ::= { prtOutputEntry 19 }
```

```
prtOutputPageDeliveryOrientation OBJECT-TYPE
-- This value is a type 1 enumeration
SYNTAX      PrtOutputPageDeliveryOrientationTC
MAX-ACCESS  read-write
MIN-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The reading surface that will be 'up' when pages are
    delivered to the associated output sub-unit. Values are
    faceUp and faceDown. (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)."
 ::= { prtOutputEntry 20 }
```

```
prtOutputBursting OBJECT-TYPE
SYNTAX      PresentOnOff
MAX-ACCESS  read-write
MIN-ACCESS  read-only
STATUS      current
DESCRIPTION
    "This object indicates that the outputting 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."
 ::= { prtOutputEntry 21 }
```

```
prtOutputDecollating OBJECT-TYPE
SYNTAX      PresentOnOff
MAX-ACCESS  read-write
MIN-ACCESS  read-only
STATUS      current
DESCRIPTION
    "This object indicates that the output supports
    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."
 ::= { prtOutputEntry 22 }
```

```
prtOutputPageCollated OBJECT-TYPE
SYNTAX      PresentOnOff
MAX-ACCESS  read-write
MIN-ACCESS  read-only
STATUS      current
DESCRIPTION
    "This object indicates that the output sub-unit
    supports page collation, and if so, whether the feature is
```

enabled. See glossary for definition of how this document defines collation."

::= { prtOutputEntry 23 }

prtOutputOffsetStacking OBJECT-TYPE

SYNTAX PresentOnOff

MAX-ACCESS read-write

MIN-ACCESS read-only

STATUS current

DESCRIPTION

"This object indicates that the output supports supports offset stacking, and if so, whether the feature is enabled. See glossary for how Offset Stacking is defined by this document."

::= { prtOutputEntry 24 }

-- The Marker Group

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

--
-- Implementation of every object in this group is mandatory.

prtMarker OBJECT IDENTIFIER ::= { printmib 10 }

prtMarkerDefaultIndex OBJECT-TYPE

SYNTAX Integer32 (1..65535)

MAX-ACCESS read-write

MIN-ACCESS read-only

STATUS current

DESCRIPTION

"The value of prtMarkerIndex corresponding to the default marker sub-unit; that is, this object selects the default marker."

::= { prtGeneralEntry 8 }

-- 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.

prtMarkerTable OBJECT-TYPE

SYNTAX SEQUENCE OF PrtMarkerEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

" "

::= { prtMarker 2 }

prtMarkerEntry OBJECT-TYPE

SYNTAX PrtMarkerEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Entries may exist in the table for each device index with a device type of 'printer'."

```
INDEX { hrDeviceIndex, prtMarkerIndex }  
::= { prtMarkerTable 1 }
```

```
PrtMarkerEntry ::= SEQUENCE {  
    prtMarkerIndex          Integer32,  
    prtMarkerMarkTech      PrtMarkerMarkTechTC,  
    prtMarkerCounterUnit   PrtMarkerCounterUnitTC,  
    prtMarkerLifeCount     Counter32,  
    prtMarkerPowerOnCount  Counter32,  
    prtMarkerProcessColorants Integer32,  
    prtMarkerSpotColorants Integer32,  
    prtMarkerAddressabilityUnit INTEGER,  
    prtMarkerAddressabilityFeedDir Integer32,  
    prtMarkerAddressabilityXFeedDir Integer32,  
    prtMarkerNorthMargin   Integer32,  
    prtMarkerSouthMargin   Integer32,  
    prtMarkerWestMargin    Integer32,  
    prtMarkerEastMargin    Integer32,  
    prtMarkerStatus        PrtSubUnitStatusTC  
}
```

```
prtMarkerIndex OBJECT-TYPE  
    SYNTAX      Integer32 (1..65535)  
    MAX-ACCESS not-accessible  
    STATUS      current  
    DESCRIPTION  
        "A unique value used by the printer to identify this marking  
        SubUnit. 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."  
    ::= { prtMarkerEntry 1 }
```

```
prtMarkerMarkTech OBJECT-TYPE  
    -- This value is a type 2 enumeration  
    SYNTAX      PrtMarkerMarkTechTC  
    MAX-ACCESS read-only  
    STATUS      current  
    DESCRIPTION  
        "The type of marking technology used for this marking sub-unit."  
    ::= { prtMarkerEntry 2 }
```

```
prtMarkerCounterUnit OBJECT-TYPE  
    -- This value is a type 1 enumeration  
    SYNTAX      PrtMarkerCounterUnitTC  
    MAX-ACCESS read-only  
    STATUS      current  
    DESCRIPTION  
        "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."  
    ::= { prtMarkerEntry 3 }
```

```
prtMarkerLifeCount OBJECT-TYPE
```

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The count of the number of units of measure counted during
    the life of printer using units of measure as specified by
    prtMarkerCounterUnit."
 ::= { prtMarkerEntry 4 }
```

```
prtMarkerPowerOnCount OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "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 prtMarkerCounterUnit."
 ::= { prtMarkerEntry 5 }
```

```
prtMarkerProcessColorants OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of process colors supported by this marker. A
    process color of 1 implies monochrome. The value of this
    object and prtMarkerSpotColorants cannot both be 0. The value of
    prtMarkerProcessColorants must be 0 or greater."
 ::= { prtMarkerEntry 6 }
```

```
prtMarkerSpotColorants OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of spot colors supported by this marker. The
    value of this object and prtMarkerProcessColorants cannot
    both be 0. Must be 0 or greater."
 ::= { prtMarkerEntry 7 }
```

```
prtMarkerAddressabilityUnit OBJECT-TYPE
-- This value is a type 1 enumeration
SYNTAX      INTEGER {
                tenThousandthsOfInches(3), -- .0001
                micrometers(4)
            }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The unit of measure of distances, as applied to the marker's
    resolution."
 ::= { prtMarkerEntry 8 }
```

```
prtMarkerAddressabilityFeedDir OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The maximum number of addressable marking positions in the feed
    direction per 10000 units of measure specified by
```

```

    prtMarkerAddressabilityUnit. A value of (-1) implies 'other' or
    'infinite' while a value of (-2) implies 'unknown'."
 ::= { prtMarkerEntry 9 }

```

prtMarkerAddressabilityXFeedDir OBJECT-TYPE

```

SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The maximum number of addressable marking positions in the cross
    feed direction in 10000 units of measure specified by
    prtMarkerAddressabilityUnit. A value of (-1) implies 'other' or
    'infinite' while a value of (-2) implies 'unknown'."
 ::= { prtMarkerEntry 10 }

```

prtMarkerNorthMargin OBJECT-TYPE

```

SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The margin, in units identified by prtMarkerAddressabilityUnit,
    from the leading edge of the medium as the medium flows
    through 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."
 ::= { prtMarkerEntry 11 }

```

prtMarkerSouthMargin OBJECT-TYPE

```

SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The margin from the South edge (see prtMarkerNorthMargin)
    of the medium in units identified by
    prtMarkerAddressabilityUnit. Printing within the area bounded by
    all four margins is guaranteed for all interpreters.
    The value (-2) means unknown."
 ::= { prtMarkerEntry 12 }

```

prtMarkerWestMargin OBJECT-TYPE

```

SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The margin from the West edge (see prtMarkerNorthMargin) of the
    medium in units identified by prtMarkerAddressabilityUnit.
    Printing within the area bounded by all four margins is
    guaranteed for all interpreters. The value (-2) means
    unknown."
 ::= { prtMarkerEntry 13 }

```

prtMarkerEastMargin OBJECT-TYPE

```

SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION

```

"The margin from the East edge (see prtMarkerNorthMargin) of the medium in units identified by prtMarkerAddressabilityUnit. Printing within the area bounded by all four margins is guaranteed for all interpreters. The value (-2) means unknown."

::= { prtMarkerEntry 14 }

prtMarkerStatus OBJECT-TYPE

SYNTAX PrtSubUnitStatusTC

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The current status of this marker sub-unit."

::= { prtMarkerEntry 15 }

-- The Marker Supplies Group

--

-- This group is optional. However, to claim conformance to this

-- group, it is necessary to implement every object in the group.

prtMarkerSupplies OBJECT IDENTIFIER ::= { printmib 11 }

prtMarkerSuppliesTable OBJECT-TYPE

SYNTAX SEQUENCE OF PrtMarkerSuppliesEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A table of the marker supplies available on this printer."

::= { prtMarkerSupplies 1 }

prtMarkerSuppliesEntry OBJECT-TYPE

SYNTAX PrtMarkerSuppliesEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Attributes of a marker supply.

Entries may exist in the table for each device index with a device type of 'printer'."

INDEX { hrDeviceIndex, prtMarkerSuppliesIndex }

::= { prtMarkerSuppliesTable 1 }

PrtMarkerSuppliesEntry ::= SEQUENCE {

prtMarkerSuppliesIndex Integer32,

prtMarkerSuppliesMarkerIndex Integer32,

prtMarkerSuppliesColorantIndex Integer32,

prtMarkerSuppliesClass PrtMarkerSuppliesClassTC,

prtMarkerSuppliesType PrtMarkerSuppliesTypeTC,

prtMarkerSuppliesDescription OCTET STRING,

prtMarkerSuppliesSupplyUnit PrtMarkerSuppliesSupplyUnitTC,

prtMarkerSuppliesMaxCapacity Integer32,

prtMarkerSuppliesLevel Integer32

}

prtMarkerSuppliesIndex OBJECT-TYPE

SYNTAX Integer32 (1..65535)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"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)."
 ::= { prtMarkerSuppliesEntry 1 }

prtMarkerSuppliesMarkerIndex OBJECT-TYPE
SYNTAX Integer32 (0..65535)
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value of prtMarkerIndex corresponding to the marking sub-unit with which this marker supply sub-unit is associated."
 ::= { prtMarkerSuppliesEntry 2 }

prtMarkerSuppliesColorantIndex OBJECT-TYPE
SYNTAX Integer32 (0..65535)
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value of prtMarkerColorantIndex corresponding to the colorant with which this marker supply sub-unit is associated. This value shall be 0 if there is no colorant table."
 ::= { prtMarkerSuppliesEntry 3 }

prtMarkerSuppliesClass OBJECT-TYPE
-- This value is a type 1 enumeration
SYNTAX PrtMarkerSuppliesClassTC
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Indicates whether this supply entity represents a supply that is consumed or a receptacle that is filled."
 ::= { prtMarkerSuppliesEntry 4 }

prtMarkerSuppliesType OBJECT-TYPE
-- This value is a type 3 enumeration
SYNTAX PrtMarkerSuppliesTypeTC
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The type of this supply."
 ::= { prtMarkerSuppliesEntry 5 }

prtMarkerSuppliesDescription OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(0..255))
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The description of this supply container/receptacle in the localization specified by prtGeneralCurrentLocalization."
 ::= { prtMarkerSuppliesEntry 6 }

prtMarkerSuppliesSupplyUnit OBJECT-TYPE
-- This value is a type 1 enumeration
SYNTAX PrtMarkerSuppliesSupplyUnitTC
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"Unit of measure of this marker supply container/receptacle."
 ::= { prtMarkerSuppliesEntry 7 }

prtMarkerSuppliesMaxCapacity OBJECT-TYPE

SYNTAX Integer32
MAX-ACCESS read-write
MIN-ACCESS read-only
STATUS current
DESCRIPTION
"The maximum capacity of this supply container/receptacle expressed in prtMarkerSuppliesSupplyUnit. If this supply container/receptacle can reliably sense this value, the value is reported by the printer and is read-only; otherwise, the value may be written (by a Remote Control 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."
 ::= { prtMarkerSuppliesEntry 8 }

prtMarkerSuppliesLevel OBJECT-TYPE

SYNTAX Integer32
MAX-ACCESS read-write
MIN-ACCESS read-only
STATUS current
DESCRIPTION
"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 reported by the printer and is read-only; otherwise, the value may be written (by a Remote Control 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. A value of (-3) means that the printer knows that there is some supply/remaining space, respectively."
 ::= { prtMarkerSuppliesEntry 9 }

-- The Marker Colorant Group

--

-- This group is optional. However, to claim conformance to this
-- group, it is necessary to implement every object in the group.

prtMarkerColorant OBJECT IDENTIFIER ::= { printmib 12 }

prtMarkerColorantTable OBJECT-TYPE

SYNTAX SEQUENCE OF PrtMarkerColorantEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"A table of all of the colorants available on the printer."
 ::= { prtMarkerColorant 1 }

prtMarkerColorantEntry OBJECT-TYPE

SYNTAX PrtMarkerColorantEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Attributes of a colorant available on the printer."

Entries may exist in the table for each device index with a device type of 'printer'."
INDEX { hrDeviceIndex, prtMarkerColorantIndex }
::= { prtMarkerColorantTable 1 }

```
PrtMarkerColorantEntry ::= SEQUENCE {  
    prtMarkerColorantIndex      Integer32,  
    prtMarkerColorantMarkerIndex Integer32,  
    prtMarkerColorantRole       PrtMarkerColorantRoleTC,  
    prtMarkerColorantValue      OCTET STRING,  
    prtMarkerColorantTonality   Integer32  
}
```

```
prtMarkerColorantIndex OBJECT-TYPE  
SYNTAX      Integer32 (1..65535)  
MAX-ACCESS not-accessible  
STATUS      current  
DESCRIPTION  
    "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)."  
::= { prtMarkerColorantEntry 1 }
```

```
prtMarkerColorantMarkerIndex OBJECT-TYPE  
SYNTAX      Integer32 (0..65535)  
MAX-ACCESS read-only  
STATUS      current  
DESCRIPTION  
    "The value of prtMarkerIndex corresponding to the  
    marker sub-unit with which this colorant entry is  
    associated."  
::= { prtMarkerColorantEntry 2 }
```

```
prtMarkerColorantRole OBJECT-TYPE  
-- This value is a type 1 enumeration  
SYNTAX      PrtMarkerColorantRoleTC  
MAX-ACCESS read-only  
STATUS      current  
DESCRIPTION  
    "The role played by this colorant."  
::= { prtMarkerColorantEntry 3 }
```

```
prtMarkerColorantValue OBJECT-TYPE  
SYNTAX      OCTET STRING (SIZE(0..255))  
MAX-ACCESS read-only  
STATUS      current  
DESCRIPTION  
    "The name of the color of this  
    colorant using standardized string names from ISO 10175 (DPA)  
    and ISO 10180 (SPDL) which are:  
    other  
    unknown  
    white  
    red  
    green  
    blue  
    cyan  
    magenta"
```

yellow
black

Implementors may add additional string values. The naming conventions in ISO 9070 are recommended in order to avoid potential name clashes"

::= { prtMarkerColorantEntry 4 }

prtMarkerColorantTonality OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"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."

::= { prtMarkerColorantEntry 5 }

-- The Media Path Group

--
-- 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 OBJECT IDENTIFIER ::= { printmib 13 }

prtMediaPathDefaultIndex OBJECT-TYPE

SYNTAX Integer32 (1..65535)

MAX-ACCESS read-write

MIN-ACCESS read-only

STATUS current

DESCRIPTION

"The value of prtMediaPathIndex corresponding to the default media path; that is, the selection of the default media path."

::= { prtGeneralEntry 9 }

prtMediaPathTable OBJECT-TYPE

SYNTAX SEQUENCE OF PrtMediaPathEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

""

::= { prtMediaPath 4 }

prtMediaPathEntry OBJECT-TYPE

SYNTAX PrtMediaPathEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Entries may exist in the table for each device index with a device type of 'printer'."

INDEX { hrDeviceIndex, prtMediaPathIndex }

::= { prtMediaPathTable 1 }

PrtMediaPathEntry ::= SEQUENCE {

```
prtMediaPathIndex          Integer32,  
prtMediaPathMaxSpeedPrintUnit  PrtMediaPathMaxSpeedPrintUnitTC,  
prtMediaPathMediaSizeUnit  PrtMediaUnitTC,  
prtMediaPathMaxSpeed      Integer32,  
prtMediaPathMaxMediaFeedDir Integer32,  
prtMediaPathMaxMediaXFeedDir Integer32,  
prtMediaPathMinMediaFeedDir Integer32,  
prtMediaPathMinMediaXFeedDir Integer32,  
prtMediaPathType          PrtMediaPathTypeTC,  
prtMediaPathDescription   OCTET STRING,  
prtMediaPathStatus        PrtSubUnitStatusTC  
}
```

```
prtMediaPathIndex OBJECT-TYPE  
SYNTAX      Integer32 (1..65535)  
MAX-ACCESS not-accessible  
STATUS      current  
DESCRIPTION  
    "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."  
 ::= { prtMediaPathEntry 1 }
```

```
prtMediaPathMaxSpeedPrintUnit OBJECT-TYPE  
-- This value is a type 1 enumeration  
SYNTAX PrtMediaPathMaxSpeedPrintUnitTC  
MAX-ACCESS read-only  
STATUS      current  
DESCRIPTION  
    "The unit of measure used in specifying the speed of all media  
    paths in the printer."  
 ::= { prtMediaPathEntry 2 }
```

```
prtMediaPathMediaSizeUnit OBJECT-TYPE  
SYNTAX      PrtMediaUnitTC  
MAX-ACCESS read-only  
STATUS      current  
DESCRIPTION  
    "The units of measure of media size for use in calculating and  
    relaying dimensional values for all media paths in the printer."  
 ::= { prtMediaPathEntry 3 }
```

```
prtMediaPathMaxSpeed OBJECT-TYPE  
SYNTAX      Integer32  
MAX-ACCESS read-only  
STATUS      current  
DESCRIPTION  
    "The maximum printing speed of this media path expressed in  
    prtMediaPathMaxSpeedUnit's. A value of (-1) implies  
    'other'. "  
 ::= { prtMediaPathEntry 4 }
```

```
prtMediaPathMaxMediaFeedDir OBJECT-TYPE  
SYNTAX      Integer32  
MAX-ACCESS read-only  
STATUS      current
```

DESCRIPTION

"The maximum physical media size in the feed direction of this media path expressed in units of measure specified by PrtMediaPathMediaSizeUnit. A value of (-1) implies 'unlimited', a value of (-2) implies 'unknown'"

::= { prtMediaPathEntry 5 }

prtMediaPathMaxMediaXFeedDir OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The maximum physical media size across the feed direction of this media path expressed in units of measure specified by prtMediaPathMediaSizeUnit. A value of (-2) implies 'unknown'."

::= { prtMediaPathEntry 6 }

prtMediaPathMinMediaFeedDir OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The minimum physical media size in the feed direction of this media path expressed in units of measure specified by prtMediaPathMediaSizeUnit. A value of (-2) implies 'unknown'."

::= { prtMediaPathEntry 7 }

prtMediaPathMinMediaXFeedDir OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The minimum physical media size across the feed direction of this media path expressed in units of measure specified by prtMediaPathMediaSizeUnit. A value of (-2) implies 'unknown'."

::= { prtMediaPathEntry 8 }

prtMediaPathType OBJECT-TYPE

-- This value is a type 2 enumeration

SYNTAX PrtMediaPathTypeTC

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The type of the media path for this media path."

::= { prtMediaPathEntry 9 }

prtMediaPathDescription OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(0..255))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The manufacturer-provided description of this media path in the localization specified by prtGeneralCurrentLocalization."

::= { prtMediaPathEntry 10 }

prtMediaPathStatus OBJECT-TYPE

SYNTAX PrtSubUnitStatusTC

MAX-ACCESS read-only

STATUS current

DESCRIPTION

```
"The current status of this media path."  
 ::= { prtMediaPathEntry 11 }  
  
-- The Print Job Delivery Channel Group  
--  
-- Implementation of every object in this group is mandatory.  
  
-- Print Job Delivery Channels are independent sources of print data. Here,  
-- print data is the term used for the information that is  
-- used to construct printed pages and 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. A channel may be independently enabled (allowing  
-- print data to flow) or disabled (stopping the flow of  
-- print data). A printer may have one or more channels.  
--  
-- The Print Job Delivery Channel table describes the capabilities of the  
-- printer and not what is currently being performed by the printer.  
--  
-- Basically, the print job delivery channel abstraction describes  
-- the final processing step of getting the print data to an  
-- interpreter. It might include some level of  
-- decompression or decoding of print stream data.  
-- channel. All of these aspects are hidden in the channel  
-- abstraction.  
--  
-- There are many kinds of print job delivery channels; some of which are based  
-- on networks and others which are not. For example, a  
-- channel can be a serial (or parallel) connection; it can  
-- be a service, such as the Unix Line Printer Daemon (LPD),  
-- offering services over a network connection (interface); or  
-- it could be a disk drive into which a floppy disk with  
-- the print data is inserted. Each print job delivery channel is typically  
-- identified by the electronic path and/or service protocol  
-- used to deliver print data to a print data interpreter.  
--  
-- 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          typically SPX/IPX based channel  
-- Novell Print Server            typically SPX/IPX based channel  
-- 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. the serial or old parallel ports)  
-- and there are services which often have elaborate  
-- protocols that run over a number of protocol stacks. In  
-- the end, what is important is the delivery of print data  
-- thru the channel.  
--  
-- The print job delivery channel sub-units are represented by the  
-- Print Job Delivery Channel Group in the Model. It has a current
```

```
-- print job 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
-- Management Applications). There is also a default
-- interpreter that is to be used if an interpreter is not
-- explicitly specified using the Control Language.

--           The print job delivery channel table and its underlying structure
--
-- The first seven items in the Print Job Delivery Channel Table define the
-- "channel" itself. A channel typically depends on other
-- protocols and interfaces to provide the data that flows
-- thru the channel.
--
-- Control of a print job delivery channel is largely limited to enabling
-- or disabling the entire channel itself. It is likely, however,
-- that more control of the process of accessing print data
-- will be needed over time. Thus, the ChannelType will
-- allow type-specific data to be associated with each
-- channel (using ChannelType specific groups in a fashion
-- analogous to the media specific MIBs that are associated
-- with the IANAIfType in the Interfaces Table). As a first
-- step in this direction, each channel will identify the
-- underlying Interface on which it is based. This is the
-- eighth object in each row of the table.
```

```
-- The Print Job Delivery Channel Table
--
-- 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 OBJECT IDENTIFIER ::= { printmib 14 }
```

```
prtChannelTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PrtChannelEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        ""
    ::= { prtChannel 1 }
```

```
prtChannelEntry OBJECT-TYPE
    SYNTAX      PrtChannelEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Entries may exist in the table for each device
        index with a device type of 'printer'."
    INDEX { hrDeviceIndex, prtChannelIndex }
    ::= { prtChannelTable 1 }
```

```
PrtChannelEntry ::= SEQUENCE {
    prtChannelIndex          Integer32,
    prtChannelType          PrtChannelTypeTC,
    prtChannelProtocolVersion OCTET STRING,
    prtChannelCurrentJobCntlLangIndex Integer32,
    prtChannelDefaultPageDescLangIndex Integer32,
    prtChannelState         INTEGER,
    prtChannelIfIndex       Integer32,
    prtChannelStatus        PrtSubUnitStatusTC,
    prtChannelInformation    DisplayString
}
```

```
prtChannelIndex OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "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."
    ::= { prtChannelEntry 1 }
```

```
prtChannelType OBJECT-TYPE
    SYNTAX      PrtChannelTypeTC
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The type of this print data channel.  This
        object provides the linkage to ChannelType-specific
```

groups that may (conceptually) extend the prtChannelTable
with additional details about that channel."
 ::= { prtChannelEntry 2 }

prtChannelProtocolVersion OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(0..63))
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The version of the protocol used on this
channel. The format used for version numbering depends
on prtChannelType."
 ::= { prtChannelEntry 3 }

prtChannelCurrentJobCntlLangIndex OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-write
MIN-ACCESS read-only
STATUS current
DESCRIPTION
"The value of prtInterpreterIndex corresponding to the
Control Language 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. PDL,
PostScript, NPAP). Must be 1 or greater."
 ::= { prtChannelEntry 4 }

prtChannelDefaultPageDescLangIndex OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-write
MIN-ACCESS read-only
STATUS current
DESCRIPTION
"The value of prtInterpreterIndex corresponding to the
Page Description Language 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., PDL, PostScript Language,
auto-sense). Must be 1 or greater."
 ::= { prtChannelEntry 5 }

prtChannelState OBJECT-TYPE
-- This value is a type 1 enumeration
SYNTAX PrtChannelStateTC
MAX-ACCESS read-write
MIN-ACCESS read-only
STATUS current
DESCRIPTION
"The state of this print data channel. The value determines
whether control information and print data is allowed through
this channel or not."
 ::= { prtChannelEntry 6 }

prtChannelIfIndex OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-write
MIN-ACCESS read-only
STATUS current

DESCRIPTION

"The value of ifIndex (in the ifTable; see the interface section of MIB-2/RFC 1213) 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 interface hierarchy. A value of zero indicates that no interface is associated with this channel."

::= { prtChannelEntry 7 }

prtChannelStatus OBJECT-TYPE

SYNTAX PrtSubUnitStatusTC

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The current status of the channel."

::= { prtChannelEntry 8 }

prtChannelInformation OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..255))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Auxiliary information to allow a printing application to use the channel for data submission to the printer. An application capable of using a specific PrtChannelType should be able to use the combined information from the prtChannelInformation and other channel and interface group objects to 'bootstrap' its use of the channel. prtChannelInformation is not intended to provide a general channel description, nor to provide information that is available once the channel is in use.

The encoding and interpretation of the prtChannelInformation object is specific to channel type. The description of each PrtChannelType enum value for which prtChannelInformation is defined specifies the appropriate encoding and interpretation, including interaction with other objects. For channel types that do not specify a prtChannelInformation value, its value shall be null (0 length).

When a new PrtChannelType enumeration value is registered, its accompanying description must specify the encoding and interpretation of the prtChannelInformation value for the channel type. prtChannelInformation semantics for an existing PrtChannelType may be added or amended in the same manner as described in section 2.4.1 for type 2 enumeration values.

The prtChannelInformation specifies values for a collection of channel attributes, represented as text according to the following rules:

1. The prtChannelInformation is coded in the NVT ASCII character set. It is not affected by localization.
2. The prtChannelInformation is a list of entries representing the attribute values. Each entry consists of the following items, in order:
 - a. a keyword, composed of alphabetic characters (A-Z, a-z), that identifies a channel attribute,
 - b. an Equals Sign (=) to delimit the keyword,

- c. a data value, consisting of NVT ASCII graphics characters (codes 32-126),
- d. a Line Feed character (code 10) to delimit the data value.

No other characters shall be present.

Keywords are case-sensitive. Conventionally, keywords are capitalized (including each word of a multi-word keyword), and, since they occupy space in the prtChannelInformation, they are kept short.

3. If a channel attribute has multiple values, it is represented by multiple entries with the same keyword, each specifying one value. Otherwise, there shall be at most one entry for each attribute.
4. By default, entries may appear in any order. If there are ordering constraints for particular entries, these must be specified in their definitions.
5. The prtChannelInformation value may represent information that is not normally coded in textual form, or that is coded in a character set other than NVT ASCII. In these cases, whatever symbolic representation is conventionally used for the information should be used for encoding the prtChannelInformation. (For instance, a binary port value might be represented as a decimal number, Unicode would be represented in UTF-8 format.)
6. For each PrtChannelType for which prtChannelInformation entries are defined, the descriptive text associated with the PrtChannelType enumeration value shall specify the following information for each entry:
 - Title: Brief description phrase, e.g.: 'Port name' 'Service Name'
 - Keyword: The keyword value, eg: 'Port' or 'Service'
 - Syntax: The encoding of the entry value, if it cannot be directly represented by NVT ASCII,
 - Status: 'Mandatory', 'Optional', or 'Conditionally Mandatory'
 - Multiplicity: 'Single' or 'Multiple' to indicate whether the entry may be present multiple times,
 - Description: Description of the use of the entry, other information required to complete the definition (e.g.: ordering constraints, interactions between entries).

Applications that interpret prtChannelInformation should ignore unrecognized entries, so they are not affected if new entry types are added."

::= { prtChannelEntry 9 }

-- The Interpreter Group

--

-- 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.
--
-- Implementation of every object in this group is mandatory.

prtInterpreter OBJECT IDENTIFIER ::= { printmib 15 }

-- Interpreter Table
--
-- 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.

prtInterpreterTable OBJECT-TYPE
SYNTAX SEQUENCE OF PrtInterpreterEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"
::= { prtInterpreter 1 }

prtInterpreterEntry OBJECT-TYPE
SYNTAX PrtInterpreterEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Entries may exist in the table for each device
index with a device type of 'printer'."
INDEX { hrDeviceIndex, prtInterpreterIndex }
::= { prtInterpreterTable 1 }

PrtInterpreterEntry ::= SEQUENCE {
prtInterpreterIndex Integer32,
prtInterpreterLangFamily PrtInterpreterLangFamilyTC,
prtInterpreterLangLevel OCTET STRING,
prtInterpreterLangVersion OCTET STRING,
prtInterpreterDescription OCTET STRING,
prtInterpreterVersion OCTET STRING,
prtInterpreterDefaultOrientation INTEGER,
prtInterpreterFeedAddressability Integer32,
prtInterpreterXFeedAddressability Integer32,
prtInterpreterDefaultCharSetIn CodedCharSet,
prtInterpreterDefaultCharSetOut CodedCharSet,
prtInterpreterTwoWay INTEGER
}

prtInterpreterIndex OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"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."  
 ::= { prtInterpreterEntry 1 }  
  
prtInterpreterLangFamily OBJECT-TYPE  
  -- This value is a type 2 enumeration  
  SYNTAX      PrtInterpreterLangFamilyTC  
  MAX-ACCESS  read-only  
  STATUS      current  
  DESCRIPTION  
    "The family name of a Page Description Language (PDL) or  
    control language which this interpreter in the printer can  
    interpret or emulate. "  
 ::= { prtInterpreterEntry 2 }  
  
prtInterpreterLangLevel OBJECT-TYPE  
  SYNTAX      OCTET STRING (SIZE(0..31))  
  MAX-ACCESS  read-only  
  STATUS      current  
  DESCRIPTION  
    "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."  
 ::= { prtInterpreterEntry 3 }  
  
prtInterpreterLangVersion OBJECT-TYPE  
  SYNTAX      OCTET STRING (SIZE(0..31))  
  MAX-ACCESS  read-only  
  STATUS      current  
  DESCRIPTION  
    "The date code or version of the language which this interpreter  
    is interpreting or emulating."  
 ::= { prtInterpreterEntry 4 }  
  
prtInterpreterDescription OBJECT-TYPE  
  SYNTAX      OCTET STRING (SIZE(0..255))  
  MAX-ACCESS  read-only  
  STATUS      current  
  DESCRIPTION  
    "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."  
 ::= { prtInterpreterEntry 5 }  
  
prtInterpreterVersion OBJECT-TYPE  
  SYNTAX      OCTET STRING (SIZE(0..31))  
  MAX-ACCESS  read-only  
  STATUS      current  
  DESCRIPTION  
    "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."  
 ::= { prtInterpreterEntry 6 }  
  
prtInterpreterDefaultOrientation OBJECT-TYPE
```

```
-- This value is a type 1 enumeration
SYNTAX      PrtInterpreterDefaultOrientationTC
MAX-ACCESS  read-write
MIN-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The current orientation default for this interpreter.  This
    value may be overridden for a particular job (e.g., by a
    command in the input data stream)."
```

```
::= { prtInterpreterEntry 7 }
```

```
prtInterpreterFeedAddressability OBJECT-TYPE
```

```
SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The maximum interpreter addressability in the feed
    direction in 10000 prtMarkerAddressabilityUnits (see
    prtMarkerAddressabilityFeedDir ) for this interpreter.
    The value (-1) means other and specifically indicates
    that the sub-unit places no restrictions on this parameter."
```

```
::= { prtInterpreterEntry 8 }
```

```
prtInterpreterXFeedAddressability OBJECT-TYPE
```

```
SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The maximum interpreter addressability in the cross feed
    direction in 10000 prtMarkerAddressabilityUnits (see
    prtMarkerAddressabilityXFeedDir) for this interpreter.
    The value (-1) means other and specifically indicates
    that the sub-unit places no restrictions on this
    parameter."
```

```
::= { prtInterpreterEntry 9 }
```

```
prtInterpreterDefaultCharSetIn OBJECT-TYPE
```

```
SYNTAX      CodedCharSet
MAX-ACCESS  read-write
MIN-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The default coded character set for input octets
    encountered outside a context in which the Page
    Description Language established the interpretation
    of the octets. (Input octets are presented to the
    interpreter through a path defined in the channel group.)
    This value shall be (2) if there is no default."
```

```
::= { prtInterpreterEntry 10 }
```

```
prtInterpreterDefaultCharSetOut OBJECT-TYPE
```

```
SYNTAX      CodedCharSet
MAX-ACCESS  read-write
MIN-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The default character set for data coming from this interpreter
    through the printer's output channel (i.e. the "backchannel").
    This value shall be (2) if there is no default."
```

```
::= { prtInterpreterEntry 11 }
```

```
prtInterpreterTwoWay OBJECT-TYPE
  -- This value is a type 1 enumeration
  SYNTAX      PrtInterpreterTwoWayTC
  MAX-ACCESS  read-only
  STATUS      current
DESCRIPTION
  "Indicates whether or not this interpreter returns information
  back to the host."
 ::= { prtInterpreterEntry 12 }

-- The Console Group
--
-- 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.
--
-- Implementation of every object in this group is mandatory.

prtConsoleLocalization OBJECT-TYPE
  SYNTAX      Integer32 (1..65535)
  MAX-ACCESS  read-write
  MIN-ACCESS  read-only
  STATUS      current
DESCRIPTION
  "The value of the prtLocalizationIndex corresponding to
  the language, country, and character set to be used for the
  console. This localization applies both to the actual display
  on the console as well as the encoding of these console
  objects in management operations."
 ::= { prtGeneralEntry 10 }

prtConsoleNumberOfDisplayLines OBJECT-TYPE
  SYNTAX      Integer32 (0..65535)
  MAX-ACCESS  read-only
  STATUS      current
DESCRIPTION
  "The number of lines on the printer's physical
  display. This value is 0 if there are no lines on the
  physical display or if there is no physical display"
 ::= { prtGeneralEntry 11 }

prtConsoleNumberOfDisplayChars OBJECT-TYPE
  SYNTAX      Integer32 (0..65535)
  MAX-ACCESS  read-only
  STATUS      current
DESCRIPTION
  "The number of characters per line displayed on the physical
  display. This value is 0 if there are no lines on the
  physical display or if there is no physical display"
 ::= { prtGeneralEntry 12 }

prtConsoleDisable OBJECT-TYPE
SYNTAX      INTEGER {
                operatorConsoleEnabled (3),
                operatorConsoleDisabled (4)
              }
  MAX-ACCESS  read-write
```

```
MIN-ACCESS read-only
STATUS      current
DESCRIPTION
    "If the object prtConsoleDisable has value
    "operatorConsoleDisabled" then input is not accepted from the
    operator console. If the object prtConsoleDisable has the value
    "operatorConsoleEnabled" then input is accepted from the
    operator console.
    "
 ::= { prtGeneralEntry 13 }

-- The Display Buffer Table

prtConsoleDisplayBuffer OBJECT IDENTIFIER ::= { printmib 16 }

prtConsoleDisplayBufferTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PrtConsoleDisplayBufferEntry
    MAX-ACCESS not-accessible
    STATUS      current
    DESCRIPTION
        ""
    ::= { prtConsoleDisplayBuffer 5 }

prtConsoleDisplayBufferEntry OBJECT-TYPE
    SYNTAX      PrtConsoleDisplayBufferEntry
    MAX-ACCESS not-accessible
    STATUS      current
    DESCRIPTION
        "This table contains one entry for each physical line on
        the display.  Lines cannot be added or deleted.

        Entries may exist in the table for each device
        index with a device type of 'printer'."
    INDEX      { hrDeviceIndex, prtConsoleDisplayBufferIndex }
    ::= { prtConsoleDisplayBufferTable 1 }

PrtConsoleDisplayBufferEntry ::= SEQUENCE {
    prtConsoleDisplayBufferIndex      Integer32,
    prtConsoleDisplayBufferText       OCTET STRING
}

prtConsoleDisplayBufferIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..65535)
    MAX-ACCESS not-accessible
    STATUS      current
    DESCRIPTION
        "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 lines to the
        printer), values are expected to remain stable across
        successive printer power cycles."
    ::= { prtConsoleDisplayBufferEntry 1 }

prtConsoleDisplayBufferText OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-write
    MIN-ACCESS read-only
    STATUS      current
    DESCRIPTION
```

"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 clears the line. It is an implementation-specific matter as to whether the agent allows a line to be overwritten before it has been cleared. Printer generated strings shall be in the localization specified by prtConsoleLocalization. Management Application generated strings should be localized by the Management Application."

```
::= { prtConsoleDisplayBufferEntry 2 }
```

-- The Console Light Table

```
prtConsoleLights OBJECT IDENTIFIER ::= { printmib 17 }
```

```
prtConsoleLightTable OBJECT-TYPE  
SYNTAX SEQUENCE OF PrtConsoleLightEntry  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION  
"  
"  
 ::= { prtConsoleLights 6 }
```

```
prtConsoleLightEntry OBJECT-TYPE  
SYNTAX PrtConsoleLightEntry  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION  
"Entries may exist in the table for each device  
index with a device type of 'printer'. "  
INDEX { hrDeviceIndex, prtConsoleLightIndex }  
 ::= { prtConsoleLightTable 1 }
```

```
PrtConsoleLightEntry ::= SEQUENCE {  
prtConsoleLightIndex Integer32,  
prtConsoleOnTime Integer32,  
prtConsoleOffTime Integer32,  
prtConsoleColor INTEGER,  
prtConsoleDescription OCTET STRING  
}
```

```
prtConsoleLightIndex OBJECT-TYPE  
SYNTAX Integer32 (0..65535)  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION  
"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."  
 ::= { prtConsoleLightEntry 1 }
```

```
prtConsoleOnTime OBJECT-TYPE  
SYNTAX Integer32  
MAX-ACCESS read-write  
MIN-ACCESS read-only
```

```
STATUS      current
DESCRIPTION
    "This object, in conjunction with prtConsoleOffTime, defines
    the current status of the light.  If both prtConsoleOnTime and
    prtConsoleOffTime are non-zero, the lamp is blinking and the
    values presented define the on time and off time, respectively,
    in milliseconds.  If prtConsoleOnTime is zero and
    prtConsoleOffTime is non-zero, the lamp is off.  If
    prtConsoleOffTime is zero and prtConsoleOnTime is non-zero,
    the lamp is on.  If both values are zero the status of the
    lamp is undefined."
 ::= { prtConsoleLightEntry 2 }

prtConsoleOffTime OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-write
MIN-ACCESS  read-only
STATUS      current
DESCRIPTION
    "This object, in conjunction with prtConsoleOnTime, defines
    the current status of the light.  If both prtConsoleOnTime and
    prtConsoleOffTime are non-zero, the lamp is blinking and the
    values presented define the on time and off time, respectively,
    in milliseconds.  If prtConsoleOnTime is zero and
    prtConsoleOffTime is non-zero, the lamp is off.  If
    prtConsoleOffTime is zero and prtConsoleOnTime is non-zero,
    the lamp is on.  If both values are zero the status of the
    lamp is undefined."
 ::= { prtConsoleLightEntry 3 }

prtConsoleColor OBJECT-TYPE
-- This value is a type 2 enumeration
SYNTAX      PrtConsoleColorTC
dMAX-ACCESS read-only
STATUS      current
DESCRIPTION
    "The color of this light."
 ::= { prtConsoleLightEntry 4 }

prtConsoleDescription OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE(0..255))
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The vendor description or label of this light in the
    localization specified by prtConsoleLocalization."
 ::= { prtConsoleLightEntry 5 }

-- The Alerts Group
--
-- 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.
--
-- Implementation of every object in this group is mandatory.

prtAlert OBJECT IDENTIFIER ::= { printmib 18 }

prtAlertTable OBJECT-TYPE
SYNTAX SEQUENCE OF PrtAlertEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"
::= { prtAlert 1 }

prtAlertEntry OBJECT-TYPE
SYNTAX PrtAlertEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Entries may exist in the table for each device
index with a device type of 'printer'."
INDEX { hrDeviceIndex, prtAlertIndex }
::= { prtAlertTable 1 }

PrtAlertEntry ::= SEQUENCE {
prtAlertIndex Integer32,
prtAlertSeverityLevel PrtSeverityLevelTC,
prtAlertTrainingLevel PrtAlertTrainingLevelTC,
prtAlertGroup PrtAlertGroupTC,
prtAlertGroupIndex Integer32,
prtAlertLocation Integer32,
prtAlertCode INTEGER,
prtAlertDescription OCTET STRING,
prtAlertTime TimeTicks
}

prtAlertIndex OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"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. 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. 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 change alerts have been deleted by requesting an attribute of each alert, and noting alerts as deleted when that retrieval is not possible."

::= { prtAlertEntry 1 }

prtAlertSeverityLevel OBJECT-TYPE

-- This value is a type 1 enumeration

SYNTAX PrtAlertSeverityLevelTC

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The level of severity of this alert table entry. The printer determines the severity level assigned to each entry into the table."

::= { prtAlertEntry 2 }

prtAlertTrainingLevel OBJECT-TYPE

-- This value is a type 2 enumeration

SYNTAX PrtAlertTrainingLevelTC

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"See textual convention PrtAlertTrainingLevelTC"

::= { prtAlertEntry 3 }

prtAlertGroup OBJECT-TYPE

-- This value is a type 1 enumeration

SYNTAX PrtAlertGroupTC

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"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.

Wherever possible, these enumerations match the sub-identifier that identifies the relevant table in the printmib."

::= { prtAlertEntry 4 }

prtAlertGroupIndex OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"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 prtAlertGroup and the prtAlertGroupIndex defines exactly which printer sub-unit caused the alert.; for example, Input #3, Output #2, and Marker #1.

Every object in this MIB is indexed with hrDeviceIndex and optionally, another index variable. If this other index

variable is present in the table that generated the alert, it will be used as the value for this object. Otherwise, this value shall be -1."
 ::= { prtAlertEntry 5 }

prtAlertLocation OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"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. The value (-2) indicates unknown"
 ::= { prtAlertEntry 6 }

prtAlertCode OBJECT-TYPE
-- This value is a type 2 enumeration
SYNTAX PrtAlertCodeTC
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"See associated textual convention PrtAlertCodeTC"
 ::= { prtAlertEntry 7 }

prtAlertDescription OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(0..255))
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"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."
 ::= { prtAlertEntry 8 }

prtAlertTime OBJECT-TYPE
SYNTAX TimeTicks
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value of sysUpTime at the time that this alert was generated."
 ::= { prtAlertEntry 9 }

printerV1Alert OBJECT-IDENTITY
STATUS current
DESCRIPTION
"The value of the enterprise-specific oid in an SNMPv1 trap sent signaling a critical event in the prtAlertTable."
 ::= { prtAlert 2 }

```
printerV2AlertPrefix OBJECT IDENTIFIER ::= { printerV1Alert 0 }

printerV2Alert NOTIFICATION-TYPE
  OBJECTS { prtAlertIndex, prtAlertSeverityLevel, prtAlertGroup,
            prtAlertGroupIndex, prtAlertLocation, prtAlertCode,
            hrDeviceIndex }
  STATUS current
  DESCRIPTION
    "This trap is sent whenever a critical event is added to the
    prtAlertTable."
  ::= { printerV2AlertPrefix 1 }

-- Note that the SNMPv2 to SNMPv1 translation rules dictate that the
-- preceding structure will result in SNMPv1 traps of the following
-- form:
--
-- printerAlert TRAP-TYPE
--   ENTERPRISE printerV1Alert
--   VARIABLES { prtAlertIndex, prtAlertSeverityLevel, prtAlertGroup,
--             prtAlertGroupIndex, prtAlertLocation, prtAlertCode,
--             hrDeviceIndex }
--   DESCRIPTION
--     "This trap is sent whenever a critical event is added to the
--     prtAlertTable."
--   ::= 1
--

-- Conformance Information

prtMIBConformance OBJECT IDENTIFIER ::= { printmib 2 }

-- compliance statements
prtMIBCompliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
    "The compliance statement for agents that implement the
    printer MIB."
  MODULE -- this module
  MANDATORY-GROUPS { prtGeneralGroup, prtInputGroup, prtOutputGroup,
                    prtMarkerGroup, prtMediaPathGroup,
                    prtChannelGroup, prtInterpreterGroup,
                    prtConsoleGroup, prtAlertTableGroup }

  OBJECT      prtGeneralReset
  SYNTAX      INTEGER {
                notResetting(3),
                resetToNVRAM(5)
              }
  DESCRIPTION
    "It is conformant to implement just these two states in
    this object.  Any additional states are optional."

  OBJECT      prtConsoleOnTime
  MIN-ACCESS  read-only
  DESCRIPTION
    "It is conformant to implement this object as read-only."
```

```
OBJECT      prtConsoleOffTime
MIN-ACCESS  read-only
DESCRIPTION
    "It is conformant to implement this object as read-only."
```

```
-- the prtResponsiblePartyGroup, prtExtendedInputGroup,
-- prtInputMediaGroup, prtExtendedOutputGroup,
-- prtOutputDimensionsGroup, prtOutputFeaturesGroup,
-- prtMarkerSuppliesGroup, prtMarkerColorantGroup,
-- prtAlertTimeGroup, prtAuxiliarySheetGroup, and the
-- prtInputSwitchingGroup are completely optional.
```

```
::= { prtMIBConformance 1 }
```

```
prtMIBGroups      OBJECT IDENTIFIER ::= { prtMIBConformance 2 }
```

```
prtGeneralGroup OBJECT-GROUP
  OBJECTS { prtGeneralConfigChanges, prtGeneralCurrentLocalization,
            prtGeneralReset, prtCoverDescription, prtCoverStatus,
            prtLocalizationLanguage, prtLocalizationCountry,
            prtLocalizationCharacterSet, prtStorageRefIndex,
            prtDeviceRefIndex }
  STATUS current
  DESCRIPTION
    "The general printer group."
  ::= { prtMIBGroups 1 }
```

```
prtResponsiblePartyGroup OBJECT-GROUP
  OBJECTS { prtGeneralCurrentOperator, prtGeneralServicePerson }
  STATUS current
  DESCRIPTION
    "The responsible party group contains contact information for
    humans responsible for the printer."
  ::= { prtMIBGroups 2 }
```

```
prtInputGroup OBJECT-GROUP
  OBJECTS { prtInputDefaultIndex, prtInputType, prtInputDimUnit,
            prtInputMediaDimFeedDirDeclared,
            prtInputMediaDimXFeedDirDeclared,
            prtInputMediaDimFeedDirChosen,
            prtInputMediaDimXFeedDirChosen, prtInputCapacityUnit,
            prtInputMaxCapacity, prtInputCurrentLevel,
            prtInputStatus, prtInputMediaName }
  STATUS current
  DESCRIPTION
    "The input group."
  ::= { prtMIBGroups 3 }
```

```
prtExtendedInputGroup OBJECT-GROUP
  OBJECTS { prtInputName, prtInputVendorName, prtInputModel,
            prtInputVersion, prtInputSerialNumber,
            prtInputDescription, prtInputSecurity }
  STATUS current
  DESCRIPTION
    "The extended input group."
  ::= { prtMIBGroups 4 }
```

```
prtInputMediaGroup OBJECT-GROUP
  OBJECTS { prtInputMediaWeight, prtInputMediaType,
```

```
        prtInputMediaColor, prtInputMediaFormParts }
STATUS   current
DESCRIPTION
    "The input media group."
 ::= { prtMIBGroups 5 }

prtOutputGroup OBJECT-GROUP
OBJECTS { prtOutputDefaultIndex, prtOutputType,
          prtOutputCapacityUnit, prtOutputMaxCapacity,
          prtOutputRemainingCapacity, prtOutputStatus }
STATUS   current
DESCRIPTION
    "The output group."
 ::= { prtMIBGroups 6 }

prtExtendedOutputGroup OBJECT-GROUP
OBJECTS { prtOutputName, prtOutputVendorName, prtOutputModel,
          prtOutputVersion, prtOutputSerialNumber,
          prtOutputDescription, prtOutputSecurity }
STATUS   current
DESCRIPTION
    "The extended output group."
 ::= { prtMIBGroups 7 }

prtOutputDimensionsGroup OBJECT-GROUP
OBJECTS { prtOutputDimUnit, prtOutputMaxDimFeedDir,
          prtOutputMaxDimXFeedDir, prtOutputMinDimFeedDir,
          prtOutputMinDimXFeedDir }
STATUS   current
DESCRIPTION
    "The output dimensions group"

 ::= { prtMIBGroups 8 }

prtOutputFeaturesGroup OBJECT-GROUP
OBJECTS { prtOutputStackingOrder,
          prtOutputPageDeliveryOrientation, prtOutputBursting,
          prtOutputDecollating, prtOutputPageCollated,
          prtOutputOffsetStacking }
STATUS   current
DESCRIPTION
    "The output features group."
 ::= { prtMIBGroups 9 }

prtMarkerGroup OBJECT-GROUP
OBJECTS { prtMarkerDefaultIndex, prtMarkerMarkTech,
          prtMarkerCounterUnit, prtMarkerLifeCount,
          prtMarkerPowerOnCount, prtMarkerProcessColorants,
          prtMarkerSpotColorants, prtMarkerAddressabilityUnit,
          prtMarkerAddressabilityFeedDir,
          prtMarkerAddressabilityXFeedDir, prtMarkerNorthMargin,
          prtMarkerSouthMargin, prtMarkerWestMargin,
          prtMarkerEastMargin, prtMarkerStatus }
STATUS   current
DESCRIPTION
    "The marker group."
 ::= { prtMIBGroups 10 }
```

```
prtMarkerSuppliesGroup OBJECT-GROUP
  OBJECTS { prtMarkerSuppliesMarkerIndex,
            prtMarkerSuppliesColorantIndex, prtMarkerSuppliesClass,
            prtMarkerSuppliesType, prtMarkerSuppliesDescription,
            prtMarkerSuppliesSupplyUnit,
            prtMarkerSuppliesMaxCapacity, prtMarkerSuppliesLevel }
  STATUS current
  DESCRIPTION
    "The marker supplies group."
  ::= { prtMIBGroups 11 }

prtMarkerColorantGroup OBJECT-GROUP
  OBJECTS { prtMarkerColorantMarkerIndex, prtMarkerColorantRole,
            prtMarkerColorantValue, prtMarkerColorantTonality }
  STATUS current
  DESCRIPTION
    "The marker colorant group."
  ::= { prtMIBGroups 12 }

prtMediaPathGroup OBJECT-GROUP
  OBJECTS { prtMediaPathDefaultIndex, prtMediaPathMaxSpeedPrintUnit,
            prtMediaPathMediaSizeUnit, prtMediaPathMaxSpeed,

prtMediaPathMaxMediaFeedDir,
            prtMediaPathMaxMediaXFeedDir,
            prtMediaPathMinMediaFeedDir,
            prtMediaPathMinMediaXFeedDir, prtMediaPathType,
            prtMediaPathDescription, prtMediaPathStatus}
  STATUS current
  DESCRIPTION
    "The media path group."
  ::= { prtMIBGroups 13 }

prtChannelGroup OBJECT-GROUP
  OBJECTS { prtChannelType, prtChannelProtocolVersion,
            prtChannelCurrentJobCntlLangIndex,
            prtChannelDefaultPageDescLangIndex, prtChannelState,
            prtChannelIfIndex, prtChannelStatus }
  STATUS current
  DESCRIPTION
    "The channel group."
  ::= { prtMIBGroups 14 }

prtInterpreterGroup OBJECT-GROUP
  OBJECTS { prtInterpreterLangFamily, prtInterpreterLangLevel,
            prtInterpreterLangVersion, prtInterpreterDescription,
            prtInterpreterVersion, prtInterpreterDefaultOrientation,
            prtInterpreterFeedAddressability,
            prtInterpreterXFeedAddressability,
            prtInterpreterDefaultCharSetIn,
            prtInterpreterDefaultCharSetOut, prtInterpreterTwoWay }
  STATUS current
  DESCRIPTION
    "The interpreter group."
  ::= { prtMIBGroups 15 }

prtConsoleGroup OBJECT-GROUP
  OBJECTS { prtConsoleLocalization, prtConsoleNumberOfDisplayLines,
            prtConsoleNumberOfDisplayChars, prtConsoleDisable,
            prtConsoleDisplayBufferText, prtConsoleOnTime,
```

```
        prtConsoleOffTime, prtConsoleColor,  
        prtConsoleDescription }  
STATUS current  
DESCRIPTION  
    "The console group."  
 ::= { prtMIBGroups 16 }  
  
prtAlertTableGroup OBJECT-GROUP  
    OBJECTS { prtAlertSeverityLevel, prtAlertTrainingLevel,  
              prtAlertGroup, prtAlertGroupIndex, prtAlertLocation,  
              prtAlertCode, prtAlertDescription }  
STATUS current  
DESCRIPTION  
    "The alert table group."  
 ::= { prtMIBGroups 17 }  
  
prtAlertTimeGroup OBJECT-GROUP  
    OBJECTS { prtAlertTime }  
STATUS current  
DESCRIPTION  
    "The alert time group."  
 ::= { prtMIBGroups 18 }  
  
prtAuxiliarySheetGroup OBJECT-GROUP  
    OBJECTS { prtGeneralStartupPage, prtGeneralBannerPage }  
STATUS current  
DESCRIPTION  
    "The auxiliary sheet group."  
 ::= { prtMIBGroups 19 }  
  
prtInputSwitchingGroup OBJECT-GROUP  
    OBJECTS { prtInputManualFeedTimeout, prtInputAutoSwitch,  
              prtInputNextIndex }  
STATUS current  
DESCRIPTION  
    "The input switching group."  
 ::= { prtMIBGroups 20 }  
  
END
```

Appendix A - Glossary of Terms

Addressability - on the marker, the number of distinctly settable marking units (pels) per unit of addressability unit; for example, 300 dots per inch is expressed as 300 per 1000 Thousandths Of Inches and 4 dots per millimeter is 4 per 1000 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

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

Binary Change Event - an event which comes 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. A binary change event may be critical or non-critical. See unary change event.

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 separate ordered sets, ready for binding.

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

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 objects that represent a type of sub-unit of the printer

IANA - Internet Assigned Numbers Authority. See STD 2, RFC 1700.

Idempotent - Idempotence is the property of an operation that results in the same state no matter how many times it is executed (at least once). This is a property that is shared by true databases in which operations on data items only change the state of the data item and do not have other side effects. Because the SNMP data model is that of operations on a database, SNMP MIB objects should be assumed to be idempotent. If a MIB object is defined in a non-idempotent way, the this data model can break in subtle ways when faced with packet loss, multiple managers, and other common conditions.

In order to fulfill the common need for actions to result from SNMP Set operations, SNMP MIB objects can be modeled such that the change in state from one state to another has the side effect of causing an action. It is important to note that with this model, an SNMP operation that sets a value equal to its current value will cause no action. This retains the idempotence of a single command, while allowing actions to be initiated by SNMP SET requests.

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

INTERNET-DRAFT

J.R. Turner, Sharp Laboratories of America

Expires June 3, 1997

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.

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.

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

Unary Change Event - an event that indicates a change of state of the printer, but to a state which is (often) just as valid as the state that was left, and from which no return is necessary. See binary change event.

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

Warning - a non-critical alert. See non-critical alert.

Appendix B - Media Size Names from ISO/IEC 10175 Document Printing Architecture

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 has no knowledge of these names, however; all media sizes in the MIB are given in terms of media dimensions as the values of `prtMediaDimFeedDir` 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 envelope size: 10 by 13 inches
na-9x12-envelope	North American 9x12 envelope size: 9 by 12 inches
na-number-10-envelope	North American number 10 business envelope size: 4.125 by 9.5 inches
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 envelope size: 10 by 14 inches
na-number-9-envelope	North American number 9 business envelope
na-6x9-envelope	North American 6x9 envelope size: 6 by 9 inches
na-10x15-envelope	North American 10x15 envelope size: 10 by 15 inches
a	engineering A size 8.5 inches by 11 inches
b	engineering B size 11 inches by 17 inches
c	engineering C size 17 inches by 22 inches
d	engineering D size 22 inches by 34 inches
e	engineering E size 34 inches by 44 inches
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-c0	ISO C0	size:	917 mm by 1297 mm
iso-c1	ISO C1	size:	648 mm by 917 mm
iso-c2	ISO C2	size:	458 mm by 648 mm
iso-c3	ISO C3	size:	324 mm by 458 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:	114 mm by 162 mm
iso-c7	ISO C7	size:	81 mm by 114 mm
iso-c8	ISO C8	size:	57 mm by 81 mm
iso-designated	ISO Designated Long		
		size:	110 mm by 220 mm
jis-b0	JIS B0	size	1030 mm by 1456 mm
jis-b1	JIS B1	size	728 mm by 1030 mm
jis-b2	JIS B2	size	515 mm by 728 mm
jis-b3	JIS B3	size	364 mm by 515 mm
jis-b4	JIS B4	size	257 mm by 364 mm
jis-b5	JIS B5	size	182 mm by 257 mm
jis-b6	JIS B6	size	128 mm by 182 mm
jis-b7	JIS B7	size	91 mm by 128 mm
jis-b8	JIS B8	size	64 mm by 91 mm
jis-b9	JIS B9	size	45 mm by 64 mm
jis-b10	JIS B10	size	32 mm by 45 mm

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 managers 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 object (prtInputDeclared-MediaDimFeedDir, prtInputDeclaredMediaDimXFeedDir, prtInputMediaType and prtInputMediaColor) in the MIB. The 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
other	
unknown	
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: 257 mm by 364 mm as defined in JIS P0138
jis-b4-coloured	Specifies the JIS B4 coloured medium with size: 257 mm by 364 mm as defined in JIS P0138
jis-b5-white	Specifies the JIS B5 white medium with size: 182 mm by 257 mm as defined in JIS P0138
jis-b5-coloured	Specifies the JIS B5 coloured medium with size: 182 mm by 257 mm as defined in JIS P0138

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 i
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 with size:
1030 mm by 1456 mm

jis-b1-white Specifies the JIS B1 white medium with size:
728 mm by 1030 mm

jis-b2-white Specifies the JIS B2 white medium with size:
515 mm by 728 mm

jis-b3-white Specifies the JIS B3 white medium with size:
364 mm by 515 mm

jis-b6-white Specifies the JIS B6 white medium with size:
257 mm by 364 mm

jis-b7-white Specifies the JIS B7 white medium with size:
182 mm by 257 mm

jis-b8-white Specifies the JIS B8 white medium with size:
128 mm by 182 mm

jis-b9-white Specifies the JIS B9 white medium with size:
91 mm by 128 mm

jis-b10-white Specifies the JIS B10 white medium with size:
64 mm by 91 mm

The following standard values are defined for engineering media:

- a Specifies the engineering A size medium with size:
8.5 inches by 11 inches
- b Specifies the engineering B size medium with size:
11 inches by 17 inches
- c Specifies the engineering C size medium with size:
17 inches by 22 inches
- d Specifies the engineering D size medium with size:
22 inches by 34 inches
- e Specifies the engineering E size medium with size:
34 inches by 44 inches

Appendix D - Roles of Users

Background

The need for "Role Models" stemmed in large part from the need to understand the importance of any given proposed object for the MIB. Many times the real world need for a proposed object would be debated within the group; the debate would typically result in the need to describe the potential usage of the object in terms of a "live" person performing some type of printing-related task.

Determining the value of a proposed object through identification of the associated human users was found to be so common that a more formalized model was required for consistent analysis. The model describing categories of human-oriented tasks is called "Role Models" in this document.

In developing the Role Models it was necessary to identify the common, primary tasks that humans typically face when interacting with a printer and its related printing system(s). It was expected that certain kinds of tasks would serve to identify the various Role Models.

In presenting the set of Role Models, the set of "Common Print System Tasks" are first presented, followed by the set of Role Model definitions. Finally, a simple matrix is presented in which Role Models and Tasks are cross-compared.

Common Print System Tasks

Upon researching the many tasks encountered by humans in dealing with printers and printing systems, the following were found to be pervasive within any operating environment:

- 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 identity 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 "tasks" cover a large spectrum of requirements surrounding the operation of a printer in a network environment. This list serves as the basis for defining the various Role Models described below.

Proposed Role Models

Following is the list of "Role Models" used to evaluate the requirements for any given Printer MIB object. 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, clearing simple paper jams, 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 organization's 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 for the purpose of planning printer acquisitions and/or placement of printers based on usage trends.

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

Accountant (ACCT) - A person responsible for tracking the usage of printing resources on an ongoing basis for the purpose of charging Users for resources used.

Matrix of Common Print System Tasks and Role Models

To better understand the relationship between the set of defined "Common Print System Tasks" and the various "Role Models," the following matrix is provided.

It is important to recognize that many of the tasks will appear to be applicable to many of the Role Models. However, when considering the actual context of a task, it is very important to realize that often the actual context of a task is such that 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 the "User" and "Operator" Role Models.

Role Models

Requirement Area	USER	OP	TECH	MGR	HELP	AM	CP	INST	ACCT
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		xx
Print job identification		xx		xx	xx		xx		xx
Internal printer status		xx	xx	xx					
Printer identification		xx		xx	xx	xx	xx	xx	xx
Printer location							xx		
Local system configuration				xx				xx	

Appendix E - Participants

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

- Ron Bergman - Dataproducts
- Luis Cubero - Hewlett-Packard
- Jay Cummings - Novell
- Andy Davidson - Tektronix
- Lee Farrell - Canon
- Joel Gyllenskog - Microworks
- Tom Hastings - Xerox
- Scott Isaacson - Novell
- Binnur Al-Kazily - Hewlett-Packard
- Rick Landau - Digital Equipment Corporation
- David Kellerman - Northlake Software
- Harry Lewis - IBM
- Pete Loya - Hewlett-Packard
- Jay Martin - Underscore
- Bob Pentecost - Hewlett-Packard
- Dave Roach - Unisys
- Bob Setterbo - Adobe
- Ron Smith - Texas Instruments
- Mike Timperman - Lexmark
- Randy Turner - Sharp Laboratories
- Bill Wagner - Digital Products, Inc.
- Chris Wellens - Interworking Labs
- Craig Whittle - Novell
- Don Wright - Lexmark
- Lloyd Young - Lexmark
- Atsushi Yuki - Kyocera
- Steve Zilles - Adobe

Security Considerations

Security issues are not discussed in this memo.

INTERNET-DRAFT
J.R. Turner, Sharp Laboratories of America
Expires June 3, 1997

Authors' Addresses

Randy Turner
Sharp Laboratories of America

Phone (360) 817-8456
Email: rturner@sharplabs.com

With significant contributions from the following individuals

Ronald L. Smith
Texas Instruments

Phone: (817) 774-6151
EMail: rlsmith@nb.ppd.ti.com

F.D. Wright
Lexmark International

Phone: (606) 232-4808
EMail: don@lexmark.com

Thomas N. Hastings
Xerox Corporation

Phone: (310) 333-6413
EMail: hastings@cp10.es.xerox.com

Stephen N. Zilles
Adobe Systems, Inc.

Phone: (415) 962-4766
EMail: szilles@mv.us.adobe.com

Joel Gyllenskog
Microworks, Inc.

(208) 375-1234
EMail: joelgyllen@aol.com

Binnur Al-Kazily
Hewlett-Packard, Inc.