```
-- Module:  PRINTER-TC
-- Editor:  Tom Hastings
-- File:    prttc.doc, .psr, .ps, .mib, .txt
-- Date:    October 25, 1995
-- Version: 0.1
```

This specification contains the Printer MIB textual-conventions which are a companion specification to the Printer MIB (see RFC 1759). These textual-conventions also includes the explanatory information about the Printer MIB and should be read first. It is intended that the Printer MIB textual-conventions and explanation can be updated and re-issued as needed with more textual-conventions and/or explanatory material, while the Printer MIB remains with no change, making the Printer MIB more stable.

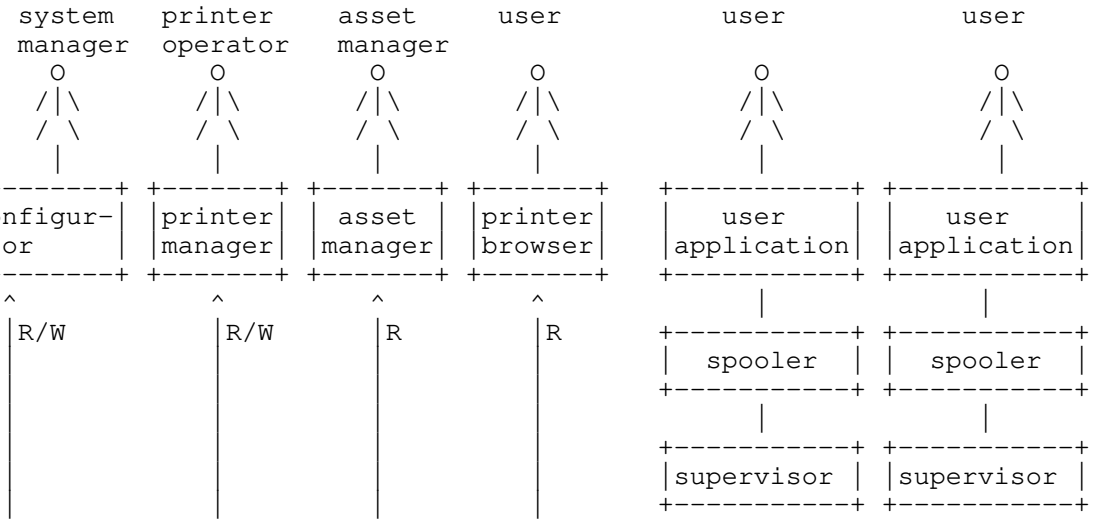
NOTE: I haven't finished copying in the enums from RFC 1759 and creating textual-convention data types for them. So far, I have only defined the textual-convention data types needed to compile the Job Monitoring MIB. However, this should give us an idea of what separating the textual-conventions into a separate module (and RFC) from the rest of the Printer MIB when we progress the Printer MIB to draft state (with two RFCs).

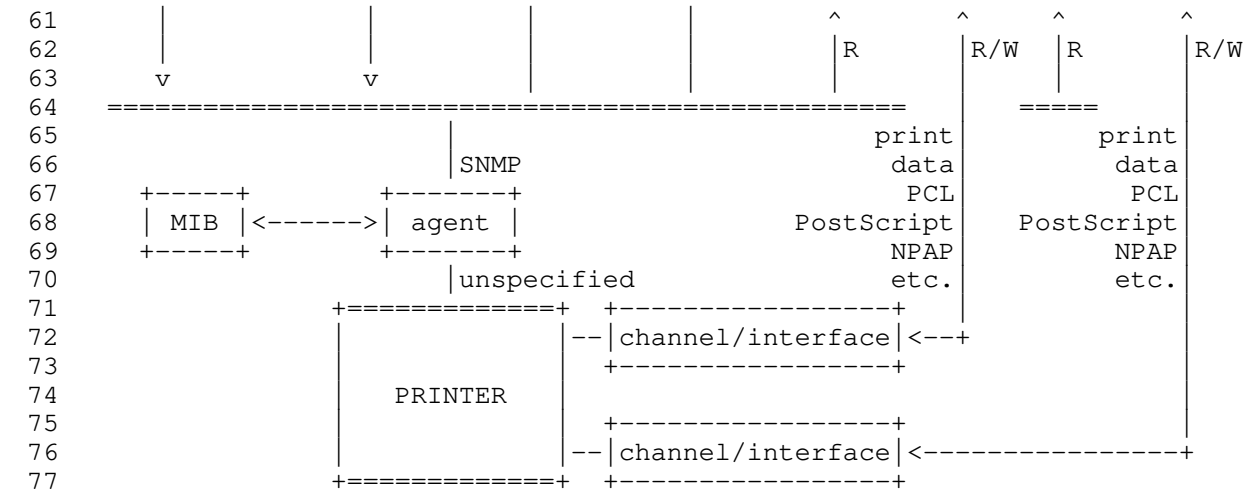
1. Introduction

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.

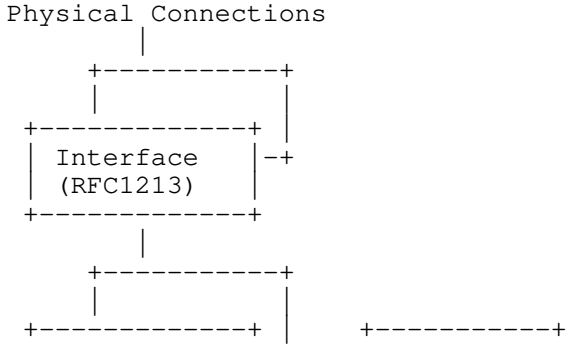
Description, status and alert information about printer can be thought of as a data base describing the printer. The management application for a printer will want to view the printer data base differently depending on how and for what purposes the information in the data base is needed.

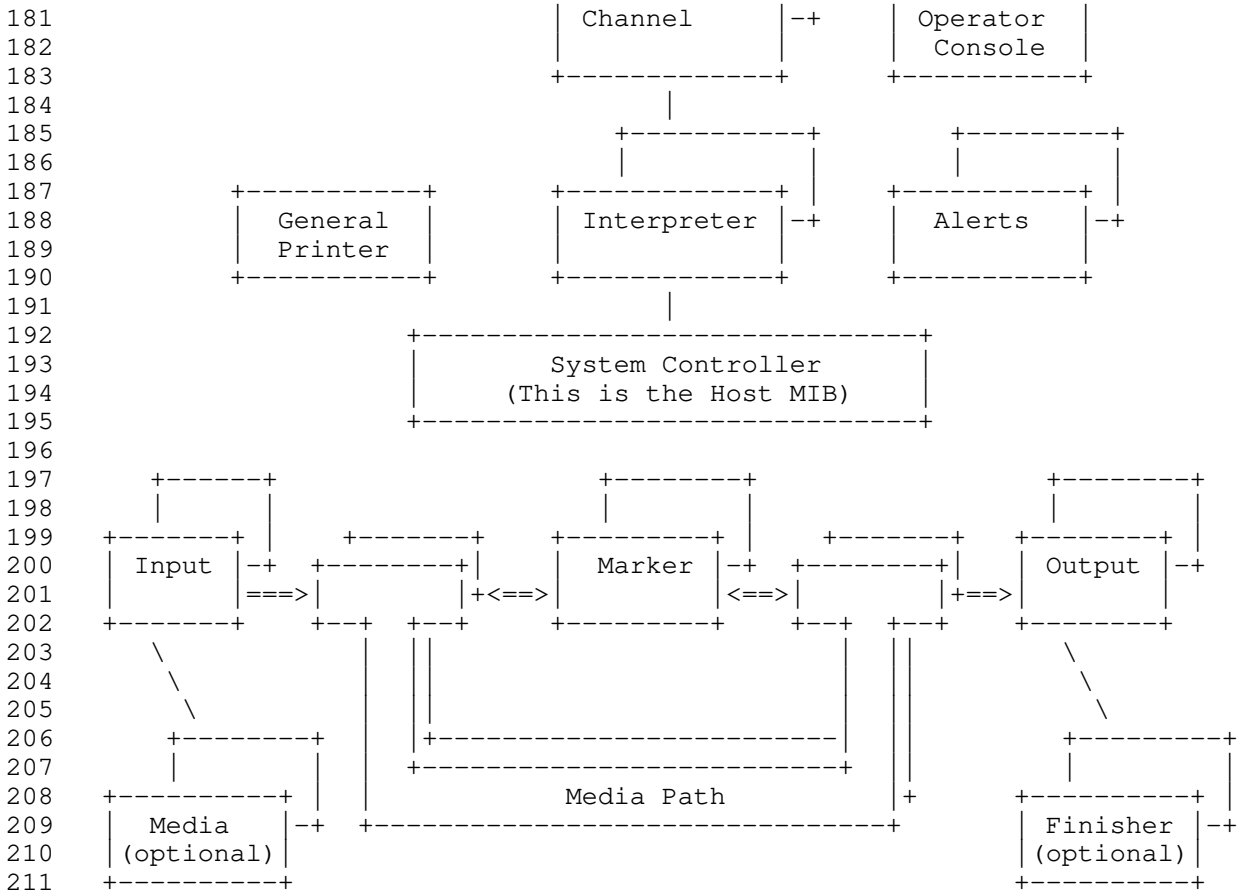
2. Printer Model

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 explicitely 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 asssociated 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 xero-graphic process, the marking process as well as the fusing process would be included in the generalized concept of the marker. With the generalized concept of a marking process, the concept of multiple marking supplies associated with a single marking sub-unit results. For example, in the xerographic process, there is not only a supply of toner, but there can also be other supplies such as a fuser supply that can be consumed and replaced separately. In addition

there can be multiple supplies of toner for a single marker device, as in a color process.

2.2.7. Media Paths

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

The channel sub-units identify the independent sources of print data (here print data is the information that is used to construct printed pages and may have both data and control aspects). A printer may have one or more channels. The channel sub-units are represented by the Channel Group in the Model. Each channel is typically identified by the electronic path and service protocol used to deliver print data to the printer. A channel sub-unit may be independently enabled (allowing print data to flow) or disabled (stopping the flow of print data). It has a current Control Language which can be used to specify which interpreter is to be used for the print data and to query and

change environment variables used by the interpreters (and SNMP). There is also a default interpreter that is to be used if an interpreter is not explicitly specified using the Control Language. Channel sub-units are based on an underlying interface.

2.2.11. Interpreters

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

2.2.12. Console

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

2.2.13. Alerts

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

2.2.13.1. Status and Alerts

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

2.2.13.2. Overall Printer Status

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.

The Host MIB provides three status objects that can be used to describe the status of a printer: (1) hrDeviceStatus in the entry in the Host MIB 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 "interesting" 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)
                         doorOpen          4        down(5)
                         jammed            5        down(5)
                         offline           6        down(5)
                         serviceRequested  7        warning(3)

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

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

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

2.2.13.2.2. Sub-unit Status

Sub-unit status is reported in the entries of the principle table in the Group that represents the sub-unit. For sub-units that report a status, there is a status column in the table and the value of this column is always 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 and overview of the table and a description of how it is managed. The basic content of the alert table is the severity (critical or non-critical) of the alert, the Group and entry where a state change caused the alert, additional information about the alert (a more detailed location, an alert code, and a description), and an indication of the level of training needed to service the alert.

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

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

2.2.13.4. Alert Table Management

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

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

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

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

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 capaciity 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 explicitely 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.

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.

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.

3. Objects from other MIB Specifications

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

3.1. System Group objects

All objects in the system group of MIB-II (RFC 1213) must be implemented.

3.2. System Controller

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

3.3. Interface Group objects

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

```
PRINTER-TC DEFINITIONS ::= BEGIN

IMPORTS
        MODULE-IDENTITY, experimental
                FROM SNMPv2-SMI                    -- RFC 1442
        TEXTUAL-CONVENTION
                FROM SNMPv2-TC;                    -- RFC 1443

  -- Upon publication as RFC, delete this comment and the line following
  -- this comment and change the reference of { printmib 102 }
  -- (below) to { mib-2 X }.
  -- This will result in changing:
  -- 1 3 6 1 3 54 printTC(102)    to:
  -- 1 3 6 1 2  1 printTC(X)
  -- This will make it easier to translate prototypes to
  -- the standard namespace because the lengths of the OID's won't
  -- change.
  printmib OBJECT IDENTIFIER ::= { experimental 54 }
```

```
printerTC MODULE-IDENTITY
        LAST-UPDATED "9510250000Z"
        ORGANIZATION "IETF/DMTF Printer Working Group (PWG)"
        CONTACT-INFO
        "       Thomas N. Hastings
                Xerox Corporation, MS ES-AE 242
                701 S. Aviation Blvd.
                El Segundo, CA 90245
                Phone:        1+ (310)333-6413
                FAX:          1+ (310)333-6342
                E-Mail:       hastings@cp10.es.xerox.com"
        DESCRIPTION
                "File: prttc.doc, .psr, .ps, .mib, .txt
                 Version: 0.1

                This textual-convention module defines textual-
                conventions for use with the Printer MIB,
                module:  Printer-MIB.  Also the explanatory
                material with this module explains the Printer
                MIB.  These textual-conventions and explanations are in
                a separate module from the Printer MIB, so that
                they may be republished when additional enums are added
                or more explanatory material is added without needing to
                republish the Printer MIB, thus increasing the
                stability of the Printer MIB."
        ::= { printmib 102 }


-- Textual conventions used in more than one group

MediaUnit ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
            "Units of measure for media dimensions.

             This is a type 1 enumeration."
    SYNTAX       INTEGER {
                     tenThousandthsOfInches(3),  -- .0001
                     micrometers(4)
                 }

CapacityUnit ::= 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)
                 }

SubUnitStatus ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
            "Status of a printer sub-unit.
```

```
1021                The SubUnitStatus is an integer that is the sum of 5
1022                distinct values, Availability, Non-Critical, Critical,
1023                On-line, and Transitioning. These values are:
1024
1025         Availability                           value
1026
1027                Available and Idle              0       000'b
1028                Available and Standby           2       010'b
1029                Available and Active            4       100'b
1030                Available and Busy              6       110'b
1031                Unavailable and OnRequest       1       001'b
1032                Unavailable because Broken      3       011'b
1033                Unknown                         5       101'b
1034
1035        Non-Critical
1036
1037                No Non-Critical Alerts          0
1038                Non-Critical Alerts             8
1039
1040        Critical
1041
1042                No Critical Alerts              0
1043                Critical Alerts                 16
1044
1045        On-Line
1046
1047                Intended state is On-Line       0
1048                Intended state is Off-Line      32
1049
1050        Transitioning
1051
1052                At intended state               0
1053                Transitioning to intended state 64
1054        "
1055
1056        SYNTAX       INTEGER (0..126)
1057
1058    PresentOnOff ::= TEXTUAL-CONVENTION
1059        STATUS       current
1060        DESCRIPTION
1061                "Presence and configuration of a device or feature.
1062
1063                 This is a type 1 enumeration."
1064        SYNTAX       INTEGER {
1065                         other(1),
1066                         on(3),
1067                         off(4),
1068                         notPresent(5)
1069                     }
1070
1071    -- General Group textual-conventions
1072
1073    PrtGeneralReset ::= TEXTUAL-CONVENTION
1074        STATUS     current
1075        DESCRIPTION
1076               "Values for reading and writing the prtGeneralReset object.
1077
1078               This value is a type 3 enumeration."
1079        SYNTAX     INTEGER {
1080                       notResetting(3),
```

```
1081                    powerCycleReset(4), -- Cold Start
1082                    resetToNVRAM(5), -- Warm Start
1083                    resetToFactoryDefaults(6) -- Reset contents of
1084                                              -- NVRAM to factory
1085                                              -- defaults
1086                }
1087
1088    CodedCharSet ::= TEXTUAL-CONVENTION
1089        STATUS     current
1090        DESCRIPTION
1091           "A coded character set value that specifies both a set of
1092            characters that may be used and an encoding (as one or more
1093            octets) that is used to represent the characters in the
1094            set. These values are to be used to identify the encoding
1095            employed for strings in the MIB where this is not fixed by
1096            the MIB.
1097
1098            Some objects that allow a choice of coded character set
1099            are: the prtLocalizationCharacterSet object in the
1100            LocalizationTable and prtInterpreterDefaultCharSetIn.
1101            The prtGeneralCurrentLocalization and prtConsoleLocalization
1102            objects in turn contain the index in the LocalizationTable
1103            of the current localization (country, language, and coded
1104            character set) of the `description' objects and the console,
1105            respectively.
1106
1107            The space of the coded character set enumeration has been
1108            divide into three regions. The first region (3-999) consists
1109            of coded character sets that have been standardized by some
1110            standard setting organization. This region is intended for
1111            standards that do not have subset implementations. The
1112            second region (1000-1999) is for the Unicode and ISO/IEC 10646
1113            coded character sets together with a specification of a (set
1114            of) sub-repetoires that may occur.  The third region (>1999)
1115            is intended for vendor specific coded character sets.
1116
1117            NOTE: Unicode and ISO 10646 character coded data may be
1118            processed and stored in either Big Endian (most significant
1119            octet first) or Little Endian (least significant octet
1120            first) order.  Intel x86, VAX, and Alpha/AXP architectures are
1121            examples of Little Endian processor architectures.
1122            Furthermore, in environments where either order may occur,
1123            so-called Unicode BYTE ORDER MARK (BOM) character (which is
1124            ISO 10646 ZERO WIDTH NO BREAK SPACE), coded as FEFF in two
1125            octets and 0000FEFF in four octets is used at the beginning
1126            of the data as a signature to indicate the order of the
1127            following data (See ISO 10646 Annex F).  Thus either
1128            ordering and BOM may occur in print data streams sent to the
1129            interpreter.  However, ISO 8824/8825 (ASN.1/BER) used by
1130            SNMP is quite clear that Big Endian order shall be used and
1131            BOM shall NOT be used in transmission in the protocol.
1132            Transmitting Unicode in Big Endian order in SNMP should
1133            not prove to be a hardship for Little Endian machines,
1134            since SNMP ASN.1/BER requires integers to be transmitted
1135            in Big Endian order as well.  So SNMP implementations on
1136            Little Endian machines are already reversing the order of
1137            integers to make them Big Endian for transmission via
1138            SNMP.  Also Unicode characters are usually treated as
1139            two-octet integers, not short text strings, so that it will
1140            be straightforward for Little Endian machines to reverse the
```

```
1141            order of Unicode character octets as well before
1142            transmitting them and after receiving them via the SNMP
1143            protocol.
1144
1145            Where a given coded character set may be known by more than
1146            one name, the most commonly known name is used as the name
1147            of the enumeration and other names are shown in the
1148            comments.  The comments also indicate where to find detailed
1149            information on the coded character set and briefly
1150            characterize its relationship to other similar coded
1151            character sets.
1152
1153            The current list of character sets and their enumerated
1154            values used to reference them is contained in the IANA
1155            Character Set registry.  The enum value is indicated by
1156            the MIBenum entry in the registry.  The enum symbol is
1157            indicated by the Alias that starts with `cs' for character
1158            set.
1159
1160            The IANA character sets registry is available via
1161            anonymous ftp.
1162
1163            The ftp server is ftp.isi.edu.
1164            The subdirectory is /in-notes/iana/assignments/.
1165            The file name is character-sets.
1166
1167            To add a character set to the IANA Registry:
1168
1169               1. Format an entry like those in the current list,
1170                  omitting the MIBenum value.
1171               2. Send the entry with a request to add the entry
1172                  to the character set list to iana@ISI.EDU.
1173               3. The IANA will supply a unique MIBenum value
1174                  and update the list.
1175
1176            This is a type 3 enumeration."
1177         SYNTAX     INTEGER {
1178           other(1)              -- used if the designated coded
1179                                 -- character set is not currently in
1180                                 -- the enumeration
1181
1182         -- See IANA Registry for standard character sets in the
1183         -- MIBenum range of 3-999.
1184
1185         -- See IANA Registry for Unicode and vendor-supplied
1186         -- combinations of ISO collections and character sets based
1187         -- on Unicode in the MIBenum range of 1000-1999.
1188         -- See IANA Registry for vendor developed character sets
1189         -- in the MIBenum range of 2000-xxxx.
1190      }
1191
1192   -- Channel Group textual-conventions
1193
1194   PrtChannelType ::= TEXTUAL-CONVENTION
1195       STATUS     current
1196       DESCRIPTION
1197           "This enumeration indicates the type of channel that is
1198           receiving jobs.
1199
1200           This is a type 2 enum."
```

```
1201        SYNTAX      INTEGER {
1202                        other(1),
1203                        chSerialPort(3),
1204                        chParallelPort(4),
1205                        chIEEE1284Port(5),
1206                        chSCSIPort(6),
1207                        chAppleTalkPAP(7),      -- AppleTalk Printer Acess
1208                                                -- Protocol
1209                        chLPDServer(8),
1210                        chNetwareRPrinter(9),   -- Netware
1211                        chNetwarePServer(10),   -- Netware
1212                        chPort9100(11),
1213                        chAppSocket(12),        -- a bi-directional, LPD-like
1214                                                -- protocol using 9101 for
1215                                                -- control and 9100 for data.
1216                                                -- Adobe Systems, Inc.
1217                        chFTP(13),              -- FTP "PUT" to printer
1218                        chTFTP(14),
1219                        chDLCLLCPort(15),
1220                        chIBM3270(16),
1221                        chIBM5250(17),
1222                        chFax(18),
1223                        chIEEE1394(19),
1224                        chTransport1(20),         -- port 35
1225                        chCPAP(21),               -- port 170
1226                        chDCERemoteProcCall(22), -- OSF
1227                        chONCRemoteProcCall(23), -- Sun Microsystems
1228                        chOLE(24),                -- Microsoft
1229                        chNamedPipe(25),
1230                        chPCPrint(26),            -- Banyan
1231                        chServerMessageBlock(27),
1232                             -- File/Print sharing protocol used by
1233                             -- various network operating systems
1234                             -- from IBM 3Com, Microsoft and others
1235                        chDPMF(28),  -- Distributed Print Mgt. Framework, IBM
1236                        chDLLAPI(29), -- Microsoft
1237                        chVxDAPI(30), -- Microsoft
1238                        chSystemObjectManager(31), -- IBM
1239                        chDECLAT(32),               -- Digital Equipment Corp.
1240                        chNPAP(33)
1241                    }
1242
1243    -- Interpreter Group textual conventions
1244
1245    PrtInterpreterLangFamily ::= TEXTUAL-CONVENTION
1246        STATUS      current
1247        DESCRIPTION
1248            "This enumeration indicates the type of interpreter that is
1249            receiving jobs.
1250
1251            This value is a type 2 enumeration."
1252        SYNTAX      INTEGER {
1253              other(1),
1254              langPCL(3),         -- PCL.  Starting with PCL version 5,
1255                                  -- HP-GL/2 is included as part of the
1256                                  -- PCL language.
1257                                  -- PCL and HP-GL/2 are registered
1258                                  -- trademarks of Hewlett-Packard Company.
1259              langHPGL(4),        -- Hewlett-Packard Graphics Language.
1260                                  -- HP-GL is a registered trademark of
```

```
1261                                -- Hewlett-Packard Company.
1262            langPJL(5),         -- Peripheral Job Language. Appears in the
1263                                -- data stream between data intended for a
1264                                -- page description language.
1265                                -- Hewlett-Packard Co.
1266            langPS(6),          -- PostScript Language (tm)
1267                                -- Postscript - a trademark of Adobe
1268                                -- Systems Incorporated which may be
1269                                -- registered in certain jurisdictions
1270            langPSPrinter(42),  -- The PostScript Language used for
1271                                -- control (with any PDLs)
1272                                -- Adobe Systems Incorporated
1273            langIPDS(7),        -- Intelligent Printer Data Stream
1274                                -- Bi-directional print data stream for
1275                                -- documents consisting of data objects
1276                                -- (text, image, graphics, bar codes),
1277                                -- resources (fonts, overlays) and page,
1278                                -- form and finishing instructions.
1279                                -- Facilitates system level device
1280                                -- control, document tracking and error
1281                                -- recovery throughout the print process.
1282                                -- Pennant Systems, IBM
1283            langPPDS(8),        -- IBM Personal Printer Data Stream.
1284                                -- Originally called IBM ASCII, the name
1285                                -- was changed to PPDS when the Laser
1286                                -- Printer was introduced in 1989.
1287                                -- Lexmark International, Inc.
1288            langEscapeP(9),
1289            langEpson(10),
1290            langDDIF(11),       -- Digital Document Interchange Format
1291                                -- Digital Equipment Corp., Maynard MA
1292            langInterpress(12),
1293            langISO6429(13),    -- ISO 6429.  Control functions for Coded
1294                                -- Character Sets (has ASCII control
1295                                -- characters, plus additional controls for
1296                                -- character imaging devices.)
1297                                -- ISO Standard, Geneva, Switzerland
1298            langLineData(14),   -- line-data:  Lines of data as separate
1299                                -- ASCII or EBCDIC records and containing
1300                                -- no control functions (no CR, LF, HT, FF,
1301                                -- etc.). For use with traditional line
1302                                -- printers.  May use CR and/or LF to
1303                                -- delimit lines, instead of records.  See
1304                                -- ISO 10175 Document Printing Application
1305                                -- (DPA)
1306                                -- ISO standard, Geneva, Switzerland
1307            langMODCA(15),      -- Mixed Object Document Content
1308                                -- Architecture
1309                                -- Definitions that allow the composition,
1310                                -- interchange, and presentation of final
1311                                -- form documents as a collection of data
1312                                -- objects (text, image, graphics, bar
1313                                -- codes), resources (fonts, overlays) and
1314                                -- page, form and finishing instructions.
1315                                -- Pennant Systems, IBM
1316            langREGIS(16),      -- Remote Graphics Instruction Set,
1317                                -- Digital Equipment Corp., Maynard MA
1318            langSCS(17),        -- SNA Character String
1319                                -- Bi-directional print data stream for SNA
1320                                -- LU-1 mode of communications
```

```
1321                                -- IBM
1322            langSPDL(18),       -- ISO 10180 Standard Page Description
1323                                -- Language
1324                                -- ISO Standard
1325            langTEK4014(19),
1326            langPDS(20),
1327            langIGP(21),
1328            langCodeV(22),      -- Magnum Code-V, Image and printer control
1329                                -- language used to control impact/dot-
1330                                -- matrix printers.
1331                                -- QMS, Inc., Mobile AL
1332            langDSCDSE(23),     -- DSC-DSE:  Data Stream Compatible and
1333                                -- Emulation Bi-directional print data
1334                                -- stream for non-SNA (DSC) and SNA LU-3
1335                                -- 3270 controller (DSE) communications
1336                                -- IBM
1337            langWPS(24),        -- Windows Printing System, Resource based
1338                                -- command/data stream used by Microsoft At
1339                                -- Work Peripherals.
1340                                -- Developed by the Microsoft Corporation.
1341            langLN03(25),       -- Early DEC-PPL3, Digital Equipment Corp.
1342            langCCITT(26),
1343            langQUIC(27),       -- QUIC (Quality Information Code), Page
1344                                -- Description Language for laser printers.
1345                                -- Included graphics, printer control
1346                                -- capability and emulation of other well-
1347                                -- known printer .
1348                                -- QMS, Inc.
1349            langCPAP(28),       -- Common Printer Access Protocol
1350                                -- Digital Equipment Corp.
1351            langDecPPL(29),     -- Digital ANSI-Compliant Printing Protocol
1352                                -- (DEC-PPL)
1353                                -- Digital Equipment Corp.
1354            langSimpleText(30),-- simple-text:  character coded data,
1355                                -- including NUL, CR , LF, HT, and FF
1356                                -- control characters.  See ISO 10175
1357                                -- Document Printing Application (DPA)
1358                                -- ISO standard, Geneva, Switzerland
1359            langNPAP(31),       -- Network Printer Alliance Protocol
1360                                -- IEEE 1284.1
1361            langDOC(32),        -- Document Option Commands, Appears in the
1362                                -- data stream between data intended for a
1363                                -- page description .
1364                                -- QMS, Inc.
1365            langimPress(33),    -- imPRESS, Page description language
1366                                -- originally developed for the ImageServer
1367                                -- line of systems. A binary language
1368                                -- providing representations for text,
1369                                -- simple graphics (rules, lines, conic
1370                                -- sections), and some large forms (simple
1371                                -- bit-map and CCITT group 3/4 encoded).The
1372                                -- language was intended to be sent over an
1373                                -- 8-bit channel and supported early
1374                                -- document preparation languages (e.g. TeX
1375                                -- and TROFF).
1376                                -- QMS, Inc.
1377            langPinwriter(34), -- 24 wire dot matrix printer  for
1378                                -- USA, Europe, and Asia except Japan.
1379                                -- More widely used in Germany, and some
1380                                -- Asian countries than in US.
```

```
1381                                -- NEC
1382            langNPDL(35),       -- Page printer  for Japanese
1383                                -- market.
1384                                -- NEC
1385            langNEC201PL(36),   -- Serial printer language used in the
1386                                -- Japanese market.
1387                                -- NEC
1388            langAutomatic(37),  -- Automatic PDL sensing.  Automatic
1389                                -- sensing of the interpreter language
1390                                -- family by the printer examining the
1391                                -- document content.  Which actual
1392                                -- interpreter language families are sensed
1393                                -- depends on the printer implementation.
1394            langPages(38),      -- Page printer Advanced Graphic Escape Set
1395                                -- IBM Japan
1396            langLIPS(39),       -- LBP Image Processing System
1397            langTIFF(40),       -- Tagged Image File Format (Aldus)
1398            langDiagnostic(41),-- A hex dump of the input to the
1399                                -- interpreter
1400            langCaPSL(43),      -- Canon Print Systems Language
1401            langEXCL(44),       -- Extended Command Language
1402                                -- Talaris Systems Inc.
1403            langLCDS(45),       -- Line Conditioned Data Stream
1404                                -- Xerox Corporation
1405            langXES(46)         -- Xerox Escape Sequences
1406                                -- Xerox Corporation
1407                 }
1408   END
```