Printer Management Interface Working Group

Printer Management Model

Version 0.4

by

Ronald L. Smith
Texas Instruments

Stephen N. Zilles
Adobe Systems, Inc

Introduction

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.

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

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

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.

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

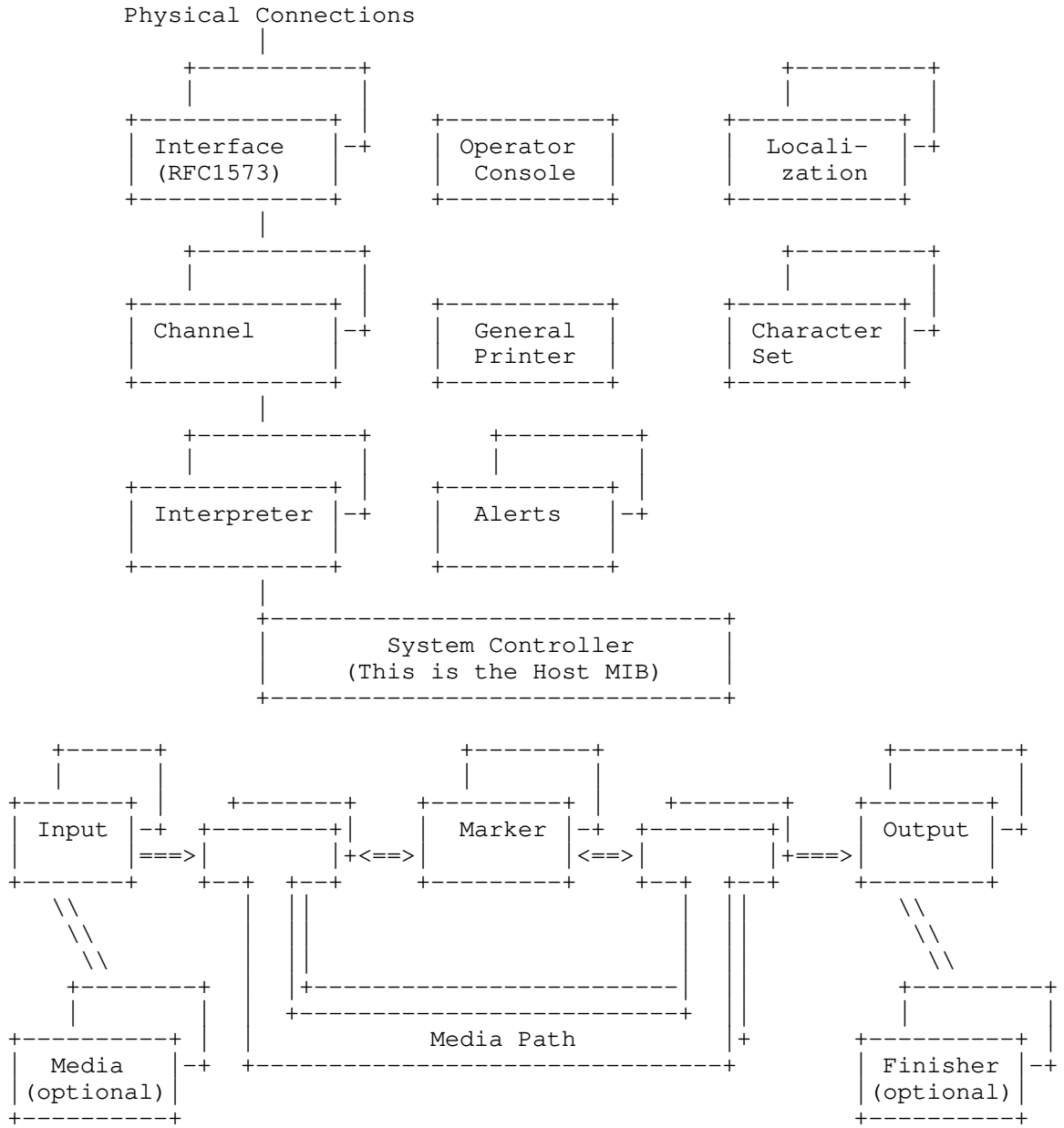
*** reference to two views, informational or problem management, was eliminated because there are more than two views ***

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

Figure 1 shows a block diagram of the printer an it's basic 16 sub-units.

Figure 1

Printer Block Diagram



An Overview of the 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 it 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.

Printer Sub-units

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.

Media

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

Outputs

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

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

Finishers

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

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

Markers

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

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

Media Paths

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

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

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.

Interfaces

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

Channels

The channel sub-units identify the independent sources of print data (here print data is the infromation 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.

Interpreters

The interpreter sub-units are responsible for the conversion of a description of intended print instances into images that is 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 PDL Interpreters and Control Language Interpreters.

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 Operator 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, 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.

General Printer

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

The localization portion of the general printer sub-unit is responsible for identifying the natural language in which messages are expressed. There may be one or more languages supported per printer. The available natural languages are represented by the Localization table.

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

Alerts

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

Status and Alerts

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

Overall Printer Status

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

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

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

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

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

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

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

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

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

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

Host MIB Printer Status

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

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

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

   hrPrinterDetectedErrorState OBJECT-TYPE
```

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

                     Condition         Bit #    hrDeviceStatus

                     lowPaper          0        warning(3)
                     noPaper           1        down(5)
                     lowToner          2        warning(3)
                     noToner           3        down(5)
                     doorOpen          4        down(5)
                     jammed            5        down(5)
                     offline           6        down(5)
                     serviceRequested  7        warning(3)

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

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

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

Sub-unit Status

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

1 other
2 unknown
3 Ready and available
4 In Critical state
5 In Non-critical, warning state
6 State is a Combination of Critical and Warning
7 On request (intervention required)
8 Unavailable
9 Busy/Temporarily Unavailable

For example, an input (tray) that jammed on the next to the last page may show a status of 5 (both a critical state, jammed, and a non critical state, low paper).

Alert Tables

The Alert Group consists of a single table in which all active alters are represented. The details of this table are given in the Management Object Summary. 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.

Alert Table Management

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

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

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

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

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

The following is a more detailed description for each of the entries in the Alert Table.

Index

The index value is used to determine which alerts have been added or removed from the alert table. This is an incrementing integer starting from zero. 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. The management application is expected to keep a local copy of the alert table. The agent will read the alert table when a trap or event notification occurs or at a periodic rate and then parse the table to determine if any new entries were added by comparing the last know index value with the current highest index value. The agent will then update its copy of the alert table.

When the printer discovers an alert is no longer active, the row for that alert will be removed from the table and the number of rows in the table will be reduced. The printer may add or delete any number of rows from the table at any time. The index value is initialized to zero each time the printer is powered on. The value is incremented each time an alert is 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 alert placed in the alert table after power up initialization of the printer will have an index value of 1.

Severity Level

This an enumeration value of the level of severity of the alert entry. The printer determines the severity level assigned to each entry into the table. The definitions of the severity levels are aligned with the ISO / DPA definitions.

Information Class

The class is an enumeration that is determined and assigned by the printer manufacturer. based on the information or the training required to handle a particular alert The printer will break alerts into three different classes. 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 three classes of alerts.

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

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

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

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

The classification of the class of an event is done by the manufacturer of the printer. Any further event classification or event class changing is the responsibility of the management layer on the system the printer is in.

Group

The sub-unit is an enumeration that defines what sub-unit type, in the printer model, that the alert is relating to. Input, output, and markers are examples of printer model sub-units.

Group Index

This is the integer designation, assigned by the printer, of the specific sub-unit within the Group that caused the alert. Combined with the Group enumeration this defines exactly which printer sub-unit cause the alert; Input 3, Output 2 and Marker 1 are examples.

Location

The sub-unit location is an integer value that is defined by the printer to further refine the location of the alert within the designated sub-unit. The location is used in conjunction with the Group enumeration and Index fields. An example is that there is an alert in Input # 2 at location number 7.

Alert Code

The code is an enumerated value that describes the type of alert for this entry in the table. There are different codes for each sub-unit type. Jam, Media Low, Toner Out and Output Full are examples of different codes for various sub-unit types. The alert codes are listed in the Management Object Summary.

Description

The description is a string that provides a description of the Alert. The description is provided by the printer to further elaborate on the enumerated alert or provide a 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.