Minutes of the Finisher MIB Project of the PWG
LA - 12/4/97

Attendance

Chuck Adams - Tektronix
Ron Bergman - Dataproducts
Paul Gloger - Xerox
Tom Hastings - Xerox
Henrik Holst - i-data International
Harry Lewis - IBM
Gary Padlipsky - Xerox
Bob Pentecost - HP
Stuart Rowley - Kyocera
Philip Thambidunai - Okidata

Requirements

1. Need to state requirements more clearly. Harry to work on this.
2. We agreed that auto-configuration of drivers is a valid requirement.

Model

Relation to Printer MIB

1. The Finisher MIB is actually an extension to the Printer MIB. As such there will be a Finisher Group added to the Printer MIB. We have to investigate the best way to do this, whether it be via an augment, a revision or what.

Finisher subUnitStatus

1. Finishing features which have entries in the Finisher group will be treated as a subUnits of the printer, and will reflect subUnit status, just as each input tray has subUnit status in the Printer MIB, today.

Path to the Finisher

1. We discussed how to describe which finishing features are accessible to which printer outputs. We considered several methods including
   a. Repeat entries in the finisher group (ex. a stapler that served printer outputs 1,2,3 would have 3, nearly identical, entries except each would include a different value in the associatedOutput index).
   b. Extend the existing prtOutputFeaturs group to include a larger list of output features than it currently supports. Then, every output entry in the Printer MIB would need a value for each of these objects, even if it only supported one finishing feature.
   c. Make use of the attribute table approach which is also being used to describe finishing feature characteristics. Thus, “outputServed” becomes a finishing feature attribute which may be multivalued if one feature serves many outputs.
   d. Have an entirely separate “path group” describing the mapping between printer outputs and finisher features.

   Status: Option c was chosen.

Minutes of the Finisher MIB Project of the PWG
LA - 12/4/97

Finisher Supplies

1. Each finisher function is a separate subunit but there will only be one supply table (like with the Marker group in the Printer MIB) with an entry for each supply. Each supply entry is indexed to one finisher subunit. If, for example, a chad collector is serving 5 finisher subunits then there will be 5 separate entries in the supply table and "remainingCapacity" will change at exactly the same rate for all 5 entries. In essence, they are "ganged".

Finisher Outputs

1. Finisher outputs are printer outputs so there is no need for a separate finisher output table.

Finisher Inputs

1. There is debate about whether to treat finisher inputs as part of the printer or not. The arguments are:
   a. Add a new prtInputType called "FinisherInput" and use this to distinguish finisher inputs from printer inputs in the Input group of the Printer MIB.
   b. The Printer MIB model shows all inputs feeding the marker. Finishers may include separate marking capabilities, but do not feed the printer marker therefore, inputs to the finisher should not be included in the printer MIB Input table.
   c. Another reason NOT to include finisher inputs with the printer MIB input table is that existing management applications may be written to display all entries in the Printer MIB Input table as printer inputs, even if they don't recognize the prtInputType enum. This would break existing applications.

   Status: Not decided. Needs to be concluded on the mailing list.

Common vs. Unique Attributes

1. Finishing features will share some common attributes which will be represented as objects in the Finisher table. The common objects are:

   Present/On?off
   Type
   CapacityUnit (can be -2)
   MaxCapacity (can be -2)
   CurrentCapacity (can be -2)
   Status
   Description
   ----- Optional----
   Vendor Name
   S/N
2. Because finishing operations are so diverse, each finishing feature will also have some unique characteristics. For example, a punch may have "shapeOfHole" as an attribute whereas a Folder may have "foldDirection". We decided to utilize a (potentially multivalued) attribute table, similar to that one used in the Job MIB, to handle this situation. All attributes of the finishing feature not addressed in the common list of objects will be covered as attributes.
3. If there is debate over what should be a common object and what should be an attribute we will attempt to apply an 80/20 rule. In other words, we won't exclude an item from object

status just because someone can articulate one example where it doesn't apply to every finishing feature. On the other hand, to be an object, the characteristic must be fairly common across finishing features.

Finishing Defaults

1. The printer MIB says what default input and output will occur. We should add defaults to the finishing group.

finSupplyTypeTC vs prtMarkerSupplyTypeTC

1. The Finisher MIB has a finSupplyTypeTC, made up of the current enumerated elements of prtMarkerSupplyTypeTC plus additional finishing supplies. The concern is about redundant registration and future conflict. Why repeat MarkerSupplies in the FinisherSuppliesTC? Because some finishers may actually include marking and/or operate with some of the same supplies.

   Status: Unresolved - needs a proposal and on-line discussion

2. Staples, Inserts and Covers were added as finSupplyTypeTCs.
3. The finisher MIF had an entire group describing the role, value and tonality of the finisher colorant. Ron has eliminated this group from the finisher MIB and, instead, simply moved finSupplyColorantValue into the finisher group as a single object.

Finishing Conflicts and Exclusions

1. We observed that some finishing features cannot be used in conjunction with other features in the same finisher. Hypothetically, you may be able to staple or fold but not staple and fold. We agreed, again, to address this problem with the attribute table, listing, for each finishing feature, the indexes of other features this one is incompatible with.

Misc.

1. The Device group disappears because it described a finisher device with multiple finisher features. Now we have a finisher group made up of printer subUnits, each associated with one or more outputs via the attribute table.
2. Finisher Feature group gets totally redone based on the new Finisher group and Attribute Table.
3. Finisher Output group disappears because finisher outputs are printer outputs.
4. Supply group remains. Do away with supplyMediaInput group and utilize the attribute table to describe the characteristics of the media.

Finishing Axis and Coordinate System

1. We spend a LOT of time discussing options for describing the corner staple operation, just in terms of how to indicate which corner to staple, not even addressing characteristics like parallel or oblique, max number to be stapled, staple size, color etc. There was a tendency to think in terms of "who had the smarts" - the finisher or the driver. I'll try to net the two main "camps" and a possible 3rd proposal which didn't have adequate time to flesh out.
   a. Reference the corners in terms of end result. Orient the reference such that the viewer is "on" the "front" side of a PORTRAIT sheet facing in a manner that would allow one to

view an image on the sheet as if it were “right-side-up”. Also, the staple would enter the paper from “above”. Then label the corners, counterclockwise starting in the reference UpperLeft as 1. (You could also say, defining the described reference orientation as “facing north”, the UpperLeft corner would be NorthWest). With this method, another orientation, like Landacape rotated clockwise from the reference, would have to refer back to the root, portrait definition to indicate staple capabilities. The advantage of this approach is that it is simplest to describe and document but it might result in having to describe the capabilities of the finisher in terms of each media type. Also, this method is more in keeping with the DPA definition and there may already be applications coded to this definition. (This needs to be confirmed).

b. Reference the corners in terms of physical capability of the finisher. We discussed a real like example of a finisher that could only staple in one of two corners on the trailing edge of the paper. The sheet might be feeding long or short edge first. With this method, the reference is defined as a viewer “on” the “front” side of a the (portrait OR landscape) sheet facing in the feed direction, calling this “North”. Again, the staple would enter the paper from “above”. Again, label corners, based off this reference, in a counter clockwise fashion starting with the “Northwest” corner. The advantage of this method appeared to be that it mapped more intuitively to the reality of the device although the abstract mapping to end result is more difficult to describe.

c. Some hybrid approach. We weren't sure exactly what this was, but there might be some distillation between the two methods which results in a more optimal hybrid.

d. Gary Padlipsky from Xerox will submit his proposal for the “Hybrid” example to the mailing list. Everyone who has an idea about the coordinate system should submit a proposal

e. Note that our intuition about where a staple “belongs” is not same for every countries writing mode. Use picture not text.

f. We delved a bit into the Postscript x,y coordinate system and started describing corners as 0,0; 0,1 etc. This may be another approach, but we still need a z axis to indicate staple direction.

2. Issue - how do we handle the ability to staple in two positions but limited according to media like can’t staple position 2 on B size. Or what about don’t want to staple transparencies?