

1 INTERNET-DRAFT

Ron Bergman
Dataproducts Corp.
Harry Lewis
IBM Corp.
March 5, 1998

8 Printer Finishing MIB

10 <draft-ietf-printmib-finishing-00.txt>

12 Expires September 5, 1998

16 Status of this Memo

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

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

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

36 Abstract

38 This document defines a printer industry standard SNMP MIB for the
39 management of printer finishing device subunits. The finishing
40 device subunits applicable to this MIB are an integral part of the
41 Printer System. This MIB does not apply to a Finisher Device that
42 is external to a Printer System.

44 The Finisher MIB is defined as an extension of the Printer MIB
45 [PrtMIB] and it is expected that the information defined in this
46 document will be incorporated into a future update of the Printer
47 MIB.

50 TABLE OF CONTENTS

51

52 1.0 INTRODUCTION..... 2

53 1.1 Scope..... 2

54 1.2 Rational..... 3

55 2.0 TERMINOLOGY..... 3

56 3.0 FINISHER SUBUNITS INTEGRATED INTO THE PRINTER MODEL:..... 4

57 4.0 PRINTER FINISHING MIB SPECIFICATION..... 4

58 5.0 REFERENCES..... 30

59 6.0 AUTHORS..... 30

60

61

62 1 INTRODUCTION

63

64 This document describes an SNMP Management Information Base (MIB) to
 65 provide for the management of in-line post-processing in a fashion that
 66 is currently provided for printers, using the Printer MIB [PrtMIB]. The
 67 Printer Finishing MIB includes the following features:

68

- 69 - Provides the status of the finishing device.
- 70 - Manages the features and configuration of the finishing device.
- 71 - Enables and disables the finishing operations.
- 72 - Allows unsolicited status from the finishing device.

73

74

75 1.1 Scope

76

77 This document provides a robust set of finishing devices, features, and
 78 functions, based upon today's state of the art of in-line finishing.
 79 Since finishing typically accompanies higher speed network printers and
 80 copiers, in contrast to simple desktop devices, no attempt is made to
 81 limit the scope to "bare minimum". On the other hand, the Printer
 82 Finishing MIB does not duplicate the production mail preparation, custom
 83 insertion, franking, and reprints that are covered by the DMTF Large
 84 Mailing Operations standard [LMO].

85

86 Information supplied by the Printer Finishing MIB may be utilized by
 87 printer and finisher management applications engaged in monitoring
 88 status and managing configuration, and also used by print and finishing
 89 submission applications which are engaged in:

90

- 91 - print-file-level finishing operations that are applied to a
 92 complete print file,
- 93
- 94 - document-level finishing operations that are applied individually
 95 to each document in the print file,
- 96
- 97 - document-level finishing operations that are applied to a selected
 98 document in the print file.

99

100 Note that not all combinations of finishing operations are compatible.
101 Compatible combinations of finishing operations are device specific.
102
103

104 1.2 Rational

105

106 The Printer MIB [PrtMIB] is now successfully deployed in a large segment
107 of the network printer market. SNMP and/or HTTP enabled printers and
108 software management applications are growing in numbers.
109

110 There is an increase in the availability of network printers and copiers
111 that include in-line finishing operations. Thus a well defined and
112 ordered set of finishing objects is now necessary for printer
113 management.
114

115 The printer model defined in the Printer MIB includes finishing
116 operations and the MIB was designed to later incorporate finisher
117 objects or to be referenced by a future Finisher MIB.
118
119
120

121 2 TERMINOLOGY

122

123 Where appropriate, the Printer Finishing MIB will conform to the
124 terminology, syntax, and semantics from the DMTF Large Mailing
125 Operations standard [LMO], the Internet Printing Protocol [IPP], and/or
126 the ISO Document Printing Application [DPA].
127

128 Finisher Input: An input tray on the finisher and not otherwise
129 associated with the printer. An example of a finisher input is a tray
130 that holds finishing "inserts".
131

132 Finisher Output: The output of the finisher. Because processing is in-
133 line, the finisher outputs are a direct extension of the set of printer
134 outputs.
135

136 Finishing Operation Axis: Defined by DPA as the axis to which some
137 finishing operations are applied to or referenced from. An example is
138 the axis for a fold operation.
139

140 Finishing Axis Offset: The offset from a finishing operation axis at
141 which the finishing operation takes place or is applied.
142
143
144

145 3 FINISHER SUBUNITS INTEGRATED INTO THE PRINTER MODEL:

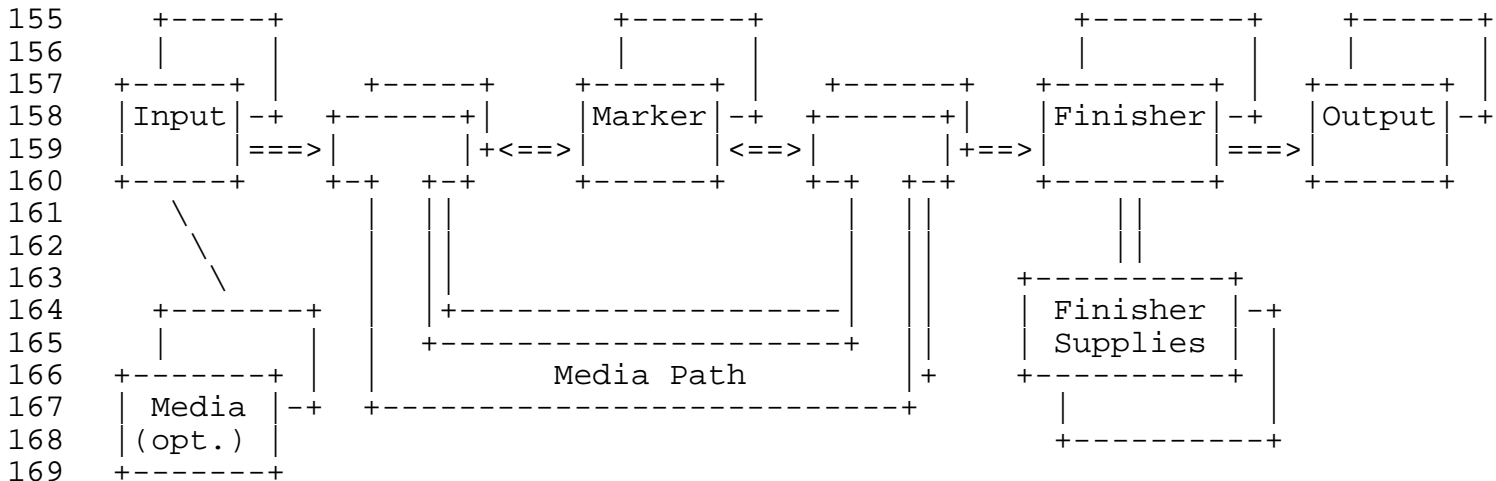
146

147 The Printer Finisher Device subunits receive media from one or more
 148 Printer Media Path subunits and deliver the media to one or more Printer
 149 Output subunits after the completion of the finishing operations. The
 150 Printer Model, as described in the Printer MIB [PRTMIB], is modified
 151 adding the finisher subunit(s) and finisher supplies as follows:

152

153

154



170

171 ISSUE: I made the finisher use the input and marker supplies tables in
 172 the Printer MIB, so that the Finisher Supplies isn't really a separate
 173 subunit. Ok? If so, then shouldn't we remove the Finisher Supplies box
 174 above?

175

176 4 THE ATTRIBUTE MECHANISM

177

178 Attributes are similar to information objects, except that attributes
 179 are identified by an enum, instead of an OID, so that attributes may be
 180 registered without requiring a new MIB. Also an implementation that
 181 does not have the functionality represented by the attribute can omit
 182 the attribute entirely, rather than having to return a distinguished
 183 value. The agent is free to materialize an attribute in the
 184 finDeviceAttributeTable as soon as the agent is aware of the value of
 185 the attribute.

186

187 The agent materializes finishing subunit attributes in a four-indexed
 188 finDeviceAttributeTable:

- 189 1. hrDeviceIndex - which device in the host
- 190 2. finDeviceIndex - which finisher subunit in the printer device
- 191 3. finDeviceAttributeTypeIndex - which attribute

192 4. finDeviceAttributeInstanceIndex - which attribute instance for
193 those attributes that can have multiple values per finishing
194 subunit.
195

196 4.1.1 Conformance of Attribute Implementation

197 An agent SHALL implement any attribute if (1) the device supports the
198 functionality represented by the attribute and (2) the information is
199 available to the agent. The agent MAY create the attribute row in the
200 finDeviceAttributeTable when the information is available or MAY create
201 the row earlier with the designated 'unknown' value appropriate for that
202 attribute. See next section.
203

204 If the server or device does not implement or does not provide access to
205 the information about an attribute, the agent SHOULD NOT create the
206 corresponding row in the finDeviceAttributeTable.

207 4.1.2 Useful, 'Unknown', and 'Other' Values for Objects and 208 Attributes

209 Some attributes have a 'useful' Integer32 value, some have a 'useful'
210 OCTET STRING value, some MAY have either or both depending on
211 implementation, and some MUST have both. See the FinAttributeTypeTC
212 textual convention for the specification of each attribute.
213 SNMP requires that if an object cannot be implemented because its values
214 cannot be accessed, then a compliant agent SHALL return an SNMP error in
215 SNMPv1 or an exception value in SNMPv2. However, this MIB has been
216 designed so that 'all' objects can and SHALL be implemented by an agent,
217 so that neither the SNMPv1 error nor the SNMPv2 exception value SHALL be
218 generated by the agent. This MIB has also been designed so that when an
219 agent materializes an attribute, the agent SHALL materialize a row
220 consisting of both the finDeviceAttributeValueAsInteger and
221 finDeviceAttributeValueAsOctets objects.
222

223 In general, values for objects and attributes have been chosen so that a
224 management application will be able to determine whether a 'useful',
225 'unknown', or 'other' value is available. When a useful value is not
226 available for an object that agent SHALL return a zero-length string for
227 octet strings, the value 'unknown(2)' for enums, a '0' value for an
228 object that represents an index in another table, and a value '-2' for
229 counting integers.
230

231 Since each attribute is represented by a row consisting of both the
232 finDeviceAttributeValueAsInteger and finDeviceAttributeValueAsOctets
233 MANDATORY objects, SNMP requires that the agent SHALL always create an
234 attribute row with both objects specified. However, for most attributes
235 the agent SHALL return a "useful" value for one of the objects and SHALL
236 return the 'other' value for the other object. For integer-only

237 attributes, the agent SHALL always return a zero-length string value for
238 the finDeviceAttributeValueAsOctets object. For octet string only
239 attributes, the agent SHALL always return a '-1' value for the
240 finDeviceAttributeValueAsInteger object.

241 4.1.3 Data Sub-types and Attribute Naming Conventions

242 Many attributes are sub-typed to give a more specific data type than
243 Integer32 or OCTET STRING. The data sub-type of each attribute is
244 indicated on the first line(s) of the description. Some attributes have
245 several different data sub-type representations. When an attribute has
246 both an Integer32 data sub-type and an OCTET STRING data sub-type, the
247 attribute can be represented in a single row in the
248 finDeviceAttributeTable. In this case, the data sub-type name is not
249 included as the last part of the name of the attribute. When the data
250 sub-types cannot be represented by a single row in the
251 finDeviceAttributeTable, each such representation is considered a
252 separate attribute and is assigned a separate name and enum value. For
253 these attributes, the name of the data sub-type is the last part of the
254 name of the attribute. For example, deviceAssociatedSupplyIndex(20) is
255 an index.

256 4.1.4 Single-Value (Row) Versus Multi-Value (MULTI-ROW) Attributes

257 Most attributes SHALL have only one row per finishing subunit. However,
258 a few attributes can have multiple values per finishing subunit, where
259 each value is a separate row in the finDeviceAttributeTable. Unless
260 indicated with 'MULTI-ROW:' in the FinAttributeTypeTC description, an
261 agent SHALL ensure that each attribute occurs only once in the
262 finDeviceAttributeTable for a finishing subunit. Most of the 'MULTI-
263 ROW' attributes do not allow duplicate values, i.e., the agent SHALL
264 ensure that each value occurs only once for a finishing subunit. Only
265 if the specification of the 'MULTI-ROW' attribute also says "There is no
266 restriction on the same xxx occurring in multiple rows" can the agent
267 allow duplicate values to occur for a single finishing subunit.
268

269 4.1.5 Linked MULTI-ROW values

270
271 Some multi-row attributes are intended to go together. Thus a set of
272 value instances represent a single instance. For example, the puncher
273 attributes indicate the shape, max size, min size, and shape of the
274 various holes that the puncher can produce. So the 1st set of values
275 could represent one kind of hole, and the 2nd set of values a another
276 kind of hole, etc.
277

278 **4.1.6 Index Value Attributes**

279

280 A number of attributes are indexes in other tables. Such attribute
281 names end with the word 'Index'. If the agent has not (yet) assigned an
282 index value for a particular index attribute for a finishing subunit,
283 the agent SHALL either: (1) return the value 0 or (2) *not* add this
284 attribute to the finDeviceAttributeTable until the index value is
285 assigned. In the interests of brevity, the semantics for 0 is specified
286 once here and is *not* repeated for each index attribute specification and
287 a DEFVAL of 0 is indicated.

288

289 5 PRINTER FINISHING MIB SPECIFICATION

290

```
291 Finisher-MIB DEFINITIONS ::= BEGIN
292
293 IMPORTS
294     MODULE-IDENTITY, OBJECT-TYPE, experimental,
295     Integer32                                     FROM SNMPv2-SMI
296     TEXTUAL-CONVENTION                           FROM SNMPv2-TC
297     MODULE-COMPLIANCE, OBJECT-GROUP              FROM SNMPv2-CONF
298     hrDeviceIndex                                FROM HOST-RESOURCES-MIB
299     PrtSubUnitStatusTC, PrtInputTypeTC,
300     PrtMarkerSuppliesSupplyUnitTC,
301     PrtCapacityUnitTC, prtOutputIndex,
302     PrtMarkerSuppliesClassTC, PresentOnOff,
303     PrtMediaPathIndex                            FROM Printer-MIB
304
305 finisherMIB MODULE-IDENTITY
306     LAST-UPDATED "9803050000Z"
307     ORGANIZATION "IETF Printer MIB Working Group"
308     CONTACT-INFO
309         "Ron Bergman
310          Dataproducts Corp.
311          1757 Tapo Canyon Road
312          Simi Valley, CA 91063-3394
313          rbergma@dpc.com
314
315          Send comments to the printmib WG using the Finisher MIB
316          Project (FIN) Mailing List:  fin@pwg.org
317
318          For further information, access the PWG web page under 'FIN':
319          http://www.pwg.org/
320
321          Implementers of this specification are encouraged to join the
322          fin mailing list in order to participate in discussions on any
323          clarifications needed and registration proposals being reviewed
324          in order to achieve consensus."
325     DESCRIPTION
326         "The MIB module for management of printer finisher units.
327         The Finisher MIB is an extension of the Printer MIB."
328     ::= { mib-2 43 }
329
330
331 -- Textual conventions for this MIB module
332
333
334 FinDeviceTypeTC ::= TEXTUAL-CONVENTION
335 -- This is a type 2 enumeration.
336     STATUS          current
337     DESCRIPTION
338         "The defined finishing device subunit types."
339     " SYNTAX          INTEGER {
340         other(1),
341         unknown(2),
```



```

342     stitcher(3),
343     folder(4),
344     binder(5),
345     trimmer(6),
346     dieCutter(7),
347     puncher(8),
348     perforater(9),
349     slitter(10),
350     separationCutter(11),
351     imprinter(12),
352     wrapper(13),
353     bander(14)
354 }
355

```

```

356 FinAttributeTypeTC ::= TEXTUAL-CONVENTION

```

```

357     STATUS      current

```

```

358     DESCRIPTION

```

```

359     "This textual convention defines a set of enums for use in
360     the finDeviceAttributeTable. The data type tag definitions
361     'INTEGER:' or 'OCTETS', indicate if the attribute can be
362     represented using the object finDeviceAttributeAsInteger or
363     the object finDeviceAttributeAsOctets, respectively. In some
364     cases, a choice between the two data types is possible and for
365     a few attributes both objects may be required at the same time
366     to properly present the value.

```

```

367
368     The attribute types defined at the time of completion of this
369     specification are:"

```

```

370
371     finAttributeTypeIndex          Data type
372     -----
373     other(1),                      Integer32
374                                     AND/OR
375                                     OCTET STRING (SIZE(0..63))
376     INTEGER: and/or OCTETS: An attribute that is not
377     currently approved and registered.

```

```

378 -- Generic Finisher subunit attributes that apply to all

```

```

379 -- Finisher subunit types (3..):

```

```

381
382     deviceCapacityUnit(3),          PrtCapacityUnitTC
383     INTEGER: The unit of measure for specifying the capacity of
384     this finisher device subunit.

```

```

385
386     deviceMaxCapacity(4),          Integer32
387     INTEGER: The maximum capacity of this finisher device
388     subunit in deviceCapacityUnits. If the device can reliably
389     sense this value, the value is sensed by the finisher device
390     and is read-only: otherwise the value may be written by a
391     management or control console application. The value (-1)
392     means other and specifically indicates that the device

```

393 places no restrictions on this parameter. The value (-2)
394 means unknown.
395

396 deviceCurrentCapacity(5), Integer32
397 INTEGER: The current capacity of this finisher device
398 subunit in deviceCapacityUnits. If the device can reliably
399 sense this value, the value is sensed by the finisher and is
400 read-only: otherwise the value may be written by a
401 management or control console application. The value (-1)
402 means other and specifically indicates that the device
403 places no restrictions on this parameter. The value (-2)
404 means unknown.
405

406 deviceAssociatedMediaPaths(6), Integer32
407 INTEGER: Indicates the media paths which can supply media
408 for this finisher device. The value of this object is a bit
409 map with each position representing the value of a
410 prtMediaPathIndex. For a media path that can be a source
411 for this finisher device subunit, the bit position equal to
412 2 raised to the power of one less than the value of
413 prtMediaPathIndex will be set.
414

415 For example, a value of hexadecimal 21 indicates the media
416 paths with a prtMediaPathIndex of 1 or 6 will supply media
417 to this finisher device subunit.
418

419 The most significant bit position SHALL not be used, which
420 allows a maximum of 31 media paths to be defined.
421 ISSUE: Why not just have a multi-valued attribute that is
422 each prtMediaPathIndex value? Then no limit and is more
423 like all other associations, instead of this bit mask.
424

425 deviceAssociatedOutputs(7), Integer32
426 INTEGER: MULTI-VALUE: Indicates the printer output
427 subunits this finisher device subunit services. The value
428 of this object is a bit map with each position representing
429 the value of a prtOutputIndex. For an output subunit that
430 is serviced by this finisher device subunit, the bit
431 position equal to 2 raised to the power of one less than the
432 value of prtOutputIndex will be set.
433

434 For example, a value of hexadecimal 49 indicates the output
435 subunits with a prtOutputIndex of 1, 4 or 7 can be serviced
436 by this finisher device subunit.
437

438 The most significant bit position SHALL not be used, which
439 allows a maximum of 31 output subunits to be defined.
440

441 If more than 31 are needed, use the next attribute value.
442

```

443         ISSUE: Why not just have a multi-valued attribute that is
444         each prtMediaPathIndex value? Then no limit and is more
445         like all other associations, instead of this bit mask.
446
447
448         deviceDescription(8),                OCTET STRING(0..255)
449         OCTETS:  A free form text description of this device subunit
450         in the localization specified by
451         prtGeneralCurrentLocalization.
452
453         deviceName(9),                      OCTET STRING(0..63)
454         OCTETS:  The name assigned to this finisher device subunit.
455
456         deviceVendorName(10),              OCTET STRING(0..63)
457         OCTETS:  The vendor name of this finisher device subunit.
458
459         deviceModel(11),                   OCTET STRING(0..63)
460         OCTETS:  The model name of this finisher device subunit."
461
462         deviceVersion(12),                 OCTET STRING(0..63)
463         OCTETS:  The version string for this finisher device
464         subunit.
465
466         deviceSerialNumber(13),           OCTET STRING(0..63)
467         OCTETS:  The serial number assigned to this finisher device
468         subunit.
469
470
471     -- Finisher Supply Attributes
472     --
473     -- A finisher subunit may have one or more supplies associated with
474     -- it. For example a finisher may use both binding tape and
475     -- stitching wire supplies. A finisher may also have more than one
476     -- source for a given type of supply e.g. multiple supply sources of
477     -- ink for imprinters.
478     --
479     -- This finisher subunit references each marker supply row in the
480     -- Printer MIB with which this finisher subunit is associated.
481
482     deviceAssociatedSupplyIndex(20)  Integer32 (0..2147483647)
483     INTEGER: MULTI-ROW: The index in the
484     prtMarkerSuppliesTable in the Printer MIB [PRTMIB] that is
485     associated with this finisher subunit. The marker supplies
486     table contains the following objects:
487     prtMarkerSuppliesIndex           Integer32,
488     prtMarkerSuppliesMarkerIndex    Integer32,
489     prtMarkerSuppliesColorantIndex  Integer32,
490     prtMarkerSuppliesClass          PrtMarkerSuppliesClassTC,
491     prtMarkerSuppliesType           PrtMarkerSuppliesTypeTC,
492     prtMarkerSuppliesDescription    OCTET STRING,
493     prtMarkerSuppliesSupplyUnit     PrtMarkerSuppliesSupplyUnitTC,

```

```

494         prtMarkerSuppliesMaxCapacity  Integer32,
495         prtMarkerSuppliesLevel        Integer32
496
497     -- Finisher Supply Media Input Attributes
498     --
499     -- A finisher subunit may have one or more associated supply media
500     -- inputs.  Each entry in this table defines an input for a
501     -- supply media type such as inserts, covers, etc.
502     --
503     -- This finisher subunit references each input row in the Printer
504     -- MIB with which this finisher subunit is associated.
505
506     deviceAssociatedInputIndex(21)      Integer32 (0..2147483647)
507     INTEGER: MULTI-ROW: The index in the prtInputTable in the
508     Printer MIB [PRTMIB] that is associated with this finisher
509     subunit.  The input table contains the following objects:
510     prtInputIndex                      Integer32,
511     prtInputTypePrt                    PrtInputTypeTC,
512     prtInputDimUnit                    PrtMediaUnitTC,
513     prtInputMediaDimFeedDirDeclared   Integer32,
514     prtInputMediaDimXFeedDirDeclared Integer32,
515     prtInputMediaDimFeedDirChosen     Integer32,
516     prtInputMediaDimXFeedDirChosen    Integer32,
517     prtInputCapacityUnit              PrtCapacityUnitTC,
518     prtInputMaxCapacity                Integer32,
519     prtInputCurrentLeve                Integer32,
520     prtInputStatus                    PrtSubUnitStatusTC,
521     prtInputMediaName                  OCTET STRING,
522     prtInputNameOCTET                  STRING,
523     prtInputVendorName                 OCTET STRING,
524     prtInputModel                      OCTET STRING,
525     prtInputVersion                    OCTET STRING,
526     prtInputSerialNumber               OCTET STRING,
527     prtInputDescription                OCTET STRING,
528     prtInputSecurity                  PresentOnOff,
529     prtInputMediaWeight                Integer32,
530     prtInputMediaType                  OCTET STRING,
531     prtInputMediaColor                 OCTET STRING,
532     prtInputMediaFormParts             Integer32,
533     prtInputMediaLoadTimeout           Integer32,
534     prtInputNextIndex                  Integer32
535
536
537
538
539
540     -- Finisher type-specific attributes:
541
542     -- stitcher attributes (30..):
543

```

```
544         stitchingType(30),                               FinStitchingTypeTC
545             INTEGER: MULTI-VALUE: The stitching operations supported
546             by this finisher subunit.
547
548 -- folder attributes (40..):
549         foldingType(40),                                   FinFoldingTypeTC
550             INTEGER: The folding operations supported by this finisher
551             subunit.
552
553 -- binder attributes (50..):
554         bindingType(50),                                   FinBindingTypeTC
555             INTEGER: The binding operations supported by this finisher
556             subunit.
557
558 -- trimmer attributes (60..):
559
560 -- die cutter attributes (70..):
561
562 -- puncher attributes (80..) These attributes are linked, so that
563 the ith value of each attribute go together:
564         punchHoleType(80),                                 FinPunchHoleTypeTC
565             INTEGER: MULTI-VALUE: The shapes of the punched holes
566             supported by this finisher subunit.
567
568         punchHoleSizeMaxDim(81),                           Integer32
569             INTEGER: MULTI-VALUE: Defines the size of the punched hole
570             in the maximum dimension. This dimension shall always be
571             measured parallel to either the long edge or the short edge
572             of the media and the maximum dimension will always be
573             measured 90 degrees from the minimum dimension. The minimum
574             and maximum dimensions may be identical.
575
576         punchHoleSizeMinDim(82),                           Integer32
577             INTEGER: Defines the size of the punched hole in the
578             minimum dimension. This dimension shall always be measured
579             parallel to either the long edge or the short edge of the
580             media and the minimum dimension will always be measured 90
581             degrees from the maximum dimension. The minimum and maximum
582             dimensions may be identical.
583
584         punchPattern(83),                                   FinPunchPatternTC
585             INTEGER: MULTI-VALUE: Defines the hole patterns produced
586             by the punch operation.
587
588 -- perforater attributes (90..):
589
590 -- slitter attributes (100..):
591         slittingType(100),                                 FinSlittingTypeTC
592             INTEGER: The slitting operations supported by this finisher
593             subunit.
594
```

```

595     -- separation cutter attributes (110..):
596
597     -- imprinter attributes (120..):
598
599     -- wrapper attributes (130..):
600         wrappingType(130),                               FinWrappingTypeTC
601         INTEGER: Provides additional information regarding the
602         wrapping operations supported by this finishing subunit.
603
604     -- bander attributes:
605
606     SYNTAX          INTEGER{
607         other(1),
608         deviceCapacityUnit(3),
609         deviceMaxCapacity(4),
610         deviceCurrentCapacity(5),
611         deviceAssociatedMediaPaths(6),
612         deviceAssociatedOutputs(7),
613         deviceDescription(8),
614         deviceName(9),
615         deviceVendorName(10),
616         deviceModel(11),
617         deviceVersion(12),
618         deviceSerialNumber(13),
619
620         stitchingType(30),
621
622         foldingType(40),
623
624         bindingType(50),
625
626         punchHoleType(80),
627         punchHoleSizeMaxDim(81),
628         punchHoleSizeMinDim(82),
629         punchPattern(83),
630
631         slittingType(100),
632
633         wrappingType(130),
634     }
635
636     FinStitchingTypeTC ::= TEXTUAL-CONVENTION
637     -- This is a type 2 enumeration.
638     STATUS          current
639     DESCRIPTION
640         "The defined stitching type enumerations."
641     SYNTAX          INTEGER {
642         other(1),
643         unknown(2),
644         staple(3),
645         stapleTopLeft(4),

```

```
646         stapleBottomLeft(5),
647         stapleTopRight(6),
648         stapleBottomRight(7),
649         saddleStitch(8),
650         edgeStitch(9),
651         stitch(10)
652     }
653
654 FinFoldingTypeTC ::= TEXTUAL-CONVENTION
655 --This is a type 2 enumeration.
656     STATUS         current
657     DESCRIPTION
658         "The defined folding device operation enumerations."
659     SYNTAX         INTEGER {
660         other(1),
661         unknown(2),
662         zFold(3),
663         halfFold(4),
664         letterFold(5)
665     }
666
667 FinBindingTypeTC ::= TEXTUAL-CONVENTION
668 -- This is a type 2 enumeration.
669     STATUS         current
670     DESCRIPTION
671         "The defined binding type enumerations."
672     SYNTAX         INTEGER {
673         other(1),
674         unknown(2),
675         bind(3),
676         tape(4),
677         plastic(5),
678         velo(6),
679         perfect(7),
680         spiral(8)
681     }
682
683 FinPunchHoleTypeTC ::= TEXTUAL-CONVENTION
684 --This is a type 2 enumeration.
685     STATUS         current
686     DESCRIPTION
687         "The defined hole type punch operation enumerations."
688     SYNTAX         INTEGER {
689         other(1),
690         unknown(2),
691         round(3),
692         oblong(4),
693         square(5),
694         rectangular(6),
695         star(7)
696     }
```

```
697
698 FinPunchPatternTC ::= TEXTUAL-CONVENTION
699 --This is a type 2 enumeration.
700     STATUS          current
701     DESCRIPTION
702         "The defined hole pattern punch operation enumerations."
703     SYNTAX          INTEGER {
704         other(1),
705         unknown(2),
706         punch(3),          --Pattern to be defined in other
707                             --attributes
708         twoHoleUS(4),      --Letter top edge, 8.5 inch side
709         threeHoleUS(5),   --Letter/ledger left edge, 11 inch side
710         twoHoleMetric(6), --A4/A3 left edge, 297 mm side
711         fourHoleMetric(7), --A4/A3 left edge, 297 mm side
712         twentyTwoHoleUS(8), --Letter/ledger left edge, 11 inch side
713         nineteenHoleUS(9) --Letter/ledger left edge, 11 inch side
714     )
715
716 FinSlittingTypeTC ::= TEXTUAL-CONVENTION
717 -- This is a type 2 enumeration.
718     STATUS          current
719     DESCRIPTION
720         "The defined slitting type enumerations."
721     SYNTAX          INTEGER {
722         other(1),
723         unknown(2),
724         slit(3),
725         slitAndSeparate(4),
726         slitAndMerge(5)
727     }
728
729 FinWrappingTypeTC ::= TEXTUAL-CONVENTION
730 --This is a type 2 enumeration.
731     STATUS          current
732     DESCRIPTION
733         "The defined wrapping device operation enumerations."
734     SYNTAX          INTEGER {
735         other(1),
736         unknown(2),
737         wrap(3),
738         shrinkWrap(4),
739         paperWrap(5)
740     }
741
742 ISSUE:  Where is the following TC used?  Can we delete it?
743
744 FinOutputTypeTC ::= TEXTUAL-CONVENTION
745 -- This is a type 2 enumeration.
746     STATUS          current
747     DESCRIPTION
```



```

748     "The defined output type enumerations."
749     SYNTAX          INTEGER {
750         other(1),
751         unknown(2),
752         removableBin(3),
753         unremovableBin(4),
754         continuousRollDevice(5),
755         mailbox(6),
756         continuousFanFold(7),
757         conveyer(8),
758         smartCart(9)
759     }
760
761     FinSupplyTypeTC ::= TEXTUAL-CONVENTION
762     -- This is a type 2 enumeration that is an extension to the
763     -- Printer MIB textual convention PrtMarkerSuppliesTypeTC.
764
765     -- *****
766     -- ISSUE: Should this just define new enums added to the Printer MIB
767     --         or keep it separate? If separate, should the duplicates
768     --         from the Printer MIB be eliminated?
769     -- *****
770
771     STATUS          current
772     DESCRIPTION
773     "The defined finishing supply type enumerations."
774     SYNTAX          INTEGER {
775         other(1),
776         unknown(2),
777         toner(3),
778         wasteToner(4),
779         ink(5),
780         inkCartridge(6),
781         inkRibbon(7),
782         wasteInk(8),
783         opc(9),                --photo conductor
784         developer(10),
785         fuserOil(11),
786         solidWax(12),
787         ribbonWax(13),
788         wasteWax(14),
789         fuser(15),
790         coronaWire(16),
791         fuserOilWick(17),
792         cleanerUnit(18),
793         fuserCleaningPad(19),
794         transferUnit(20),
795         tonerCartridge(21),
796         fuserOiler(22),       --Supply types 3 to 22 are from the
797                                --Printer MIB
798         water(23),

```

```

799         wasteWater(24),
800         glueWaterAdditive(25),
801         wastePaper(26),
802         bindingTape(27),
803         bandingTape(28),
804         stitchingWire(29),
805         shrinkWrap(30),
806         paperWrap(31),
807         staples(32),
808         inserts(33),
809         covers(34)
810     }
811
812
813 -- Finisher Device Group (Mandatory)
814 --
815 -- A printer may support zero or more finishing subunits. A
816 -- finishing device subunit may be associated with one or more
817 -- output subunits and one or more media path subunits.
818 --
819 -- NOTE: The objects in this table could not have been made attributes
820 -- because the Printer MIB trap mechanism needs to be able to indicate
821 -- alerts in subunits which are rows in tables.
822
823 finDeviceTable OBJECT-TYPE
824     SYNTAX      SEQUENCE OF FinDeviceEntry
825     MAX-ACCESS  not-accessible
826     STATUS      current
827     DESCRIPTION
828         "This table defines the finishing device subunits,
829         including information regarding possible configuration
830         options and the status for each finisher device subunit."
831     ::= { finisherMIB 18 }
832
833 finDeviceEntry OBJECT-TYPE
834     SYNTAX      FinDeviceEntry
835     MAX-ACCESS  not-accessible
836     STATUS      current
837     DESCRIPTION
838         "There is an entry in the finishing device table for each
839         possible finisher operation."
840     INDEX { hrDeviceIndex, finDeviceIndex }
841     ::= { finDevicesTable 1 }
842
843 FinDeviceEntry ::= SEQUENCE {
844     finDeviceIndex      Integer32,
845     finDeviceType       FinDeviceTypeTC,
846     finDevicePresentOnOff PresentOnOff,
847     finDeviceStatus     PrtSubUnitStatusTC
848 }
849

```

```

850 finDeviceIndex OBJECT-TYPE
851     SYNTAX      Integer32(0..2147483647)
852     MAX-ACCESS  not-accessible
853     STATUS      current
854     DESCRIPTION
855         "A unique value used to identify a finisher operation.
856         Although these values may change due to a major
857         reconfiguration of the printer system (e.g. the addition
858         of new finishing operations), the values are normally
859         expected to remain stable across successive power cycles."
860     ::= { finDeviceEntry 1 }
861
862 finDeviceType OBJECT-TYPE
863     SYNTAX      FinDeviceTypeTC
864     MAX-ACCESS  read-only
865     STATUS      current
866     DESCRIPTION
867         "Defines the type of finishing operation associated with this
868         table row entry."
869     ::= { finDeviceEntry 2 }
870
871 finDevicePresentOnOff OBJECT-TYPE
872     SYNTAX      PresentOnOff
873     MAX-ACCESS  read-write
874     STATUS      current
875     DESCRIPTION
876         "Indicates if this finishing device subunit is available
877         and whether the device subunit is enabled."
878     ::= { finDeviceEntry 3 }
879
880 finDeviceStatus OBJECT-TYPE
881     SYNTAX      PrtSubUnitStatusTC
882     MAX-ACCESS  read-only
883     STATUS      current
884     DESCRIPTION
885         "Indicates the current status of this finisher device
886         subunit."
887     ::= { finDeviceEntry 4 }
888
889
890 "----- Finisher
891 Device Attribute Group (Mandatory)
892 --
893 -- A finisher device subunit may have one or more parameters that
894 -- cannot be specified by any other objects in the MIB. The
895 -- Device Attribute group allows the definition of these
896 -- parameters.
897
898 finDeviceAttributeTable OBJECT-TYPE
899     SYNTAX      SEQUENCE OF FinDeviceAttributeEntry
900     MAX-ACCESS  not-accessible

```

```

901     STATUS      current
902     DESCRIPTION
903         "The attribute table defines special parameters that are
904         applicable only to a minority of the finisher devices.
905         An attribute table entry is used, rather than unique
906         objects, to minimize the number of MIB objects and to
907         allow for expansion without the addition of MIB objects.
908         Each finisher device is represented by a separate row
909         in the device subunit attribute table."
910 ::= { finisherMIB 21 }
911
912 finDeviceAttributeEntry OBJECT-TYPE
913     SYNTAX      FinDeviceAttributeEntry
914     MAX-ACCESS not-accessible
915     STATUS      current
916     DESCRIPTION
917         "Each entry defines a finisher function parameter that
918         cannot be represented by an object in the finisher
919         device subunit table."
920     INDEX { hrDeviceIndex, finDeviceIndex,
921            finDeviceAttributeTypeIndex }
922 ::= { finDeviceAttributeTable 1 }
923
924 FinDeviceAttributeEntry ::= SEQUENCE {
925     finDeviceAttributeTypeIndex      FinAttributeTypeTC,
926     finDeviceAttributeInstanceIndex  Integer32,
927     finDeviceAttributeValueAsInteger Integer32,
928     finDeviceAttributeValueAsOctets  OCTET STRING
929 }
930
931 finDeviceAttributeTypeIndex OBJECT-TYPE
932     SYNTAX      FinAttributeTypeTC
933     MAX-ACCESS not-accessible
934     STATUS      current
935     DESCRIPTION
936         "Defines the attribute type represented by this row."
937 ::= { finDeviceAttributeEntry 1 }
938
939 finDeviceAttributeInstanceIndex OBJECT-TYPE
940     SYNTAX      Integer32
941     MAX-ACCESS not-accessible
942     STATUS      current
943     DESCRIPTION
944         "An index that allows the discrimination of an attribute
945         instance when the same attribute occurs multiple times for
946         a specific instance of a finisher function. The value of
947         this index shall be 1 if only a single instance of the
948         attribute occurs for the specific finisher function."
949 ::= { finDeviceAttributeEntry 2 }
950
951 finDeviceAttributeValueAsInteger OBJECT-TYPE

```

```
952     SYNTAX      Integer32
953     MAX-ACCESS  read-only
954     STATUS      current
955     DESCRIPTION
956         "Defines the integer value of the attribute.  The value of
957         the attribute is represented as an integer if the
958         FinAttributeTypeTC description for the attribute has the
959         tag 'INTEGER:'.
```

Depending upon the attribute enum definition, this object may be either an integer, a counter, an index, or an enum. Attributes for which the concept of an integer value is not meaningful SHALL return a value of -1 for this attribute."

```
966     ::= { finDeviceAttributeEntry 3 }
```

```
968 finDeviceAttributeValueAsOctets OBJECT-TYPE
```

```
969     SYNTAX      OCTET STRING (SIZE(0..63))
970     MAX-ACCESS  read-only
971     STATUS      current
972     DESCRIPTION
973         "Contains the octet string value of the attribute.  The
974         value of the attribute is represented as a string if the
975         FinAttributeTypeTC description for the attribute has the
976         tag 'OCTETS:'.
```

Depending upon the attribute enum definition, this object may be either a coded character set string (text) or a binary octet string. Attributes for which the concept of an octet string value is not meaningful SHALL contain a zero length string."

```
983     ::= { finDeviceAttributeEntry 4 }
```

```
985 END
```

990 6 REFERENCES

```
991 [PRTMIB] The Printer MIB, RFC 1759, IETF standards track document.
```

```
992 [LMO] Large Mailing Operations Specification, DMTF.  See
993 http://www.dmtf.org/tech/apps.html
```

```
994 [DPA] ISO/IEC 10175 Document Printing Application (DPA).  See
995 ftp://ftp.pwg.org/pub/pwg/dpa/
```

```
996 [IPP] Internet Printing Protocol/1.0: Model and Semantics, work
1000 in progress on the IETF standards track.  See draft-ietf-ipp-model-
```

1002 09.txt.

1003

1004

1005

1006 7 AUTHORS

1007

1008 This document was created with significant contributions from the
1009 following individuals.

1010

1011 Ron Bergman (Editor)
1012 Dataproducts Corp.
1013 1757 Tapo Canyon Road
1014 Simi Valley, CA 93063-3394

1015

1016 Phone: 805-578-4421
1017 Fax: 805-578-4001
1018 Email: rbergman@dpc.com

1019

1020

1021 Harry Lewis (chairman)
1022 IBM Corporation
1023 6300 Diagonal Hwy
1024 Boulder, CO 80301

1025

1026 Phone: (303) 924-5337
1027 Fax: (303) 924-4662
1028 Email: harryl@us.ibm.com

1029

1030

1031

1032 Send comments to the Printer Working Group (PWG) using the Finisher
1033 MIB Project (FIN) Mailing List: fin@pwg.org

1034

1035 For further information, access the PWG web page under "FIN":
1036 <http://www.pwg.org/>

1037

1038

1039 Other Participants:

1040

1041 Chuck Adams - Tektronix
1042 Andy Davidson - Tektronix
1043 Mabry Dozier - QMS
1044 Lee Ferrel - Canon
1045 Paul Gloger - Xerox
1046 Richard Hart - Digital
1047 Tom Hastings - Xerox
1048 Scott Isaacson - Novell
1049 David Kellerman - Northlake Software
1050 Henrik Holst - i-data International
1051 Rick Landau - Digital

1052 Jay Martin - Underscore
1053 Gary Padlipski - Xerox
1054 Bob Pentecost - HP
1055 Stuart Rowley - Kyocera
1056 Yuki Sacchi - Japan Computer Industry
1057 Philip Thambidunai - Okidata
1058 William Wagner - DPI/Osicom
1059 Chris Wellens - Interworking Labs
1060 Don Wright - Lexmark
1061 Lloyd Young - Lexmark
1062
1063