

1

2 **Open Standard Print API (PAPI): Additions for Printer**
3 **Capabilities API**

4

Version 0.3 (DRAFT)

5

5

6 **Open Standard Print API (PAPI): Additions for Printer Capabilities API: Version 0.3**
7 **(DRAFT)**

8 Version 0.3 (DRAFT) Edition

9 Copyright © 2002 by Free Standards Group

10 Permission to use, copy, modify and distribute this document for any purpose and without fee is hereby granted in
11 perpetuity, provided that the above copyright notice and this paragraph appear in all copies.

12 **Table of Contents**

13 **1. Printer Capabilities..... 1**

14 1.1. Introduction..... 1

15 1.2. Definitions 1

16 1.3. Objectives..... 2

17 1.3.1. Standard printer capabilities API..... 2

18 1.3.2. Independent of underlying source of capabilities 2

19 1.3.3. Support returning information in context 2

20 1.3.4. Support returning constraints 3

21 1.3.5. Support returning display hints..... 3

22 1.3.6. Support logically grouping features..... 3

23 1.4. Interface..... 3

24 1.4.1. Query Function..... 3

25 1.4.2. Capabilities Attributes..... 3

26 1.4.3. Validation Function 5

27 **A. Change History..... 6**

28 Chapter 1. Printer Capabilities

29 1.1. Introduction

30 In the context of this document, *printer capabilities* refers to information about the
31 features, options, limitations, etc. of a print device (either an actual device, or an
32 abstract device which may represent a group or pool of actual devices). This
33 includes such information as:

- 34 • Does the printer support color printing?
- 35 • At what resolution(s) can the printer print?
- 36 • What input trays are present?
- 37 • What size media is loaded in each tray?
- 38 • Which trays are manual-feed and which are auto-feed?
- 39 • Can the printer print duplex output?
- 40 • What is the printable area on each of the loaded media?
- 41 • What output bins are present?
- 42 • What finishings (staple, punch, etc.) does the printer support?
- 43 • What combinations of features are not allowed together?
- 44 • What features should be presented on the print user interface?
- 45 • ...and many others...

46 The uses of printer capabilities by applications include:

- 47 1. To control how to display print options in a print UI dialog. Examples:
 - 48 • What values to put in the bin selection pull-down list
 - 49 • Whether or not to gray-out the duplex option when a particular output bin
 - 50 has been selected
 - 51 • Whether or not to display a color vs. back-and-white selection
- 52 2. To control how the print datastream is generated. Examples:
 - 53 • How large an image to draw to fill the printable area
 - 54 • How much to shift the image if "3-hole punch" finishing has been selected
 - 55 • How to request that the printer print on paper from the manual envelope
 - 56 feeder
- 57 3. To do job validation and printer selection. Examples:
 - 58 • Can I print this job with these options on this printer?
 - 59 • Find a printer which can print this job with these options.

60 1.2. Definitions

61 **Driver:**

62 In the context of this document, this is a software program that, possibly together
63 with some external representation of printer capabilities, can translate generic
64 graphic/drawing commands issued from an application into a printer-specific
65 datastream which will render those commands on paper. The driver may also be
66 able to transform graphic/drawing commands from an input datastream into a
67 printer-specific output datastream (e.g. translate Postscript into raster images).

68 **PPD (Postscript Printer Description) files:**

69 Files which contain capabilities information in a special text format that was
70 developed by Adobe for devices which include a Postscript interpreter. In addition
71 to capabilities information, PPD files contain information about how to present
72 capabilities to an end-user (e.g. in a GUI dialog) and how features can be selected

73 and settings can be changed. Postscript drivers rely heavily on PPD files to generate
74 the correct Postscript datastream. PPD files are heavily used on both Windows and
75 Unix platforms, and on Linux they currently represent the primary repository for
76 capabilities information. The specification of the PPD format can be found at
77 http://partners.adobe.com/asn/developer/pdfs/tn/5003.PPD_Spec_v4.3.pdf.

78 **UPDF (Universal Printer Description Format):**

79 This is a relatively new, standard XML format for representing printer capabilities.
80 UPDF is not tied to a particular printer datastream such as Postscript, and it is
81 intended to support representation of dynamic printer capabilities better than PPD.

82 **Constraint:**

83 This is a restriction on the printer capabilities where some combination of two or
84 more attributes/values are not allowed together. This may be due to printer
85 hardware limitations or to the disallowing of combinations which do not make
86 sense by the printer vendor or the print system administrator. An simple example
87 constraint would be "transparencies cannot be selected when printing duplex".

88 Note: The term "constraint" is more properly used to describe the *inclusion* of a set
89 of allowed combinations of features. However, this PAPI specification follows the
90 pre-existing printing industry convention of using "constraint" to describe the
91 *exclusion* of a set of disallowed combinations of features.

92 **1.3. Objectives**

93 This section attempts to describe the objectives of the PAPI printer capabilities
94 support. It is important to understand these objectives in order to understand why
95 the support is structured the way that it is.

96 **1.3.1. Standard printer capabilities API**

97 There is no standard API which a Linux application can use to retrieve printer
98 capabilities regardless of the device, the driver, and the print server being used.
99 This makes it very difficult for application writers to support generating print data
100 without writing multiple versions of the print logic or without tying the application
101 to very specific print system environments. This specification provides the standard
102 API, making applications which use it independent of the underlying print system.

103 **1.3.2. Independent of underlying source of capabilities**

104 The capabilities information returned to the application could come from many
105 different sources and be in many different formats, including:

- 106 • PPD files
- 107 • UPDF database
- 108 • SNMP queries
- 109 • Device drivers

110 The API defined here must hide these differences so that the application is
111 independent of which of the above implementation(s) are used.

112 **1.3.3. Support returning information in context**

113 The API must support a means for requesting capabilities information *in the context*
114 *of* a particular set of job options. For example, a way is needed to request the printer
115 capabilities given that medium and color/black-and-white selections have already
116 been made.

117 **1.3.4. Support returning constraints**

118 The API must support a means for returning constraints on printer capabilities (see
 119 earlier definition of "constraint"). This allows applications to not submit jobs with
 120 disallowed combinations of options, and to display better print job dialogs (gray-
 121 out potentially conflicting options, highlight conflicting options that have been
 122 selected, display an error message when invalid combinations are submitted, etc.).

123 The constraints returned should allow some level of "boolean logic", including
 124 negation, to simplify the information returned. For example, to not allow doing
 125 finishing when transparencies are selected as the medium, it would be preferable if
 126 the constraints could express "(type = transparency) AND (finishing NOT= none)"
 127 instead of having to list a combination of "(type = transparency)" with every
 128 possible finishing value other than "none".

129 **1.3.5. Support returning display hints**

130 The API should support a means for returning "display hints". This is information
 131 that the application can use to display print options in a print dialog that is easy to
 132 use. For example, returning information about which options should be displayed
 133 on the "main window", which should be displayed in an "advanced" dialog, and
 134 which should not be displayed at all.

135 **1.3.6. Support logically grouping features**

136 The API should support a means for returning logical groupings of printer
 137 features. This is information about combinations of lower-level features that can be
 138 displayed and selected as a group to make the user interface easier to use. For
 139 example, a group of features called "black-and-white-draft" could include a logical
 140 setting of the color, resolution, and print density options.

141 The feature group support should be an open, extendible way for printer vendors
 142 and print administrators to express logical and commonly used groupings of print
 143 options that make it easier for end-users to take advantage of lower-level printer
 144 features. They should *not* be used to blindly list all possible combinations of a set of
 145 options, whether or not all the combinations make sense.

146 **1.4. Interface**

147 **1.4.1. Query Function**

148 The API used by the application to retrieve printer capabilities is the
 149 `papiPrinterQuery` function. See the description of that function for further details.

150 **1.4.2. Capabilities Attributes**

151 In addition to the xxx-supported attributes defined by the IPP standard [RFC2911],
 152 this section defines new attributes needed to satisfy the objectives described earlier.

153 **1.4.2.1. job-constraints-col (1setOf collection)**

154 Constraints are expressed in the printer object's `job-constraints-col` attribute. This
 155 attribute is multivalued with each value having collection syntax. Each value is, in
 156 fact, an attribute list that represents *one* combination of job attributes/values which
 157 are not allowed for that printer. If an attribute in the collection does not have a
 158 value, then *all* values of that attribute are disallowed in this combination.

159 The set of values associated with `job-constraints-col` represents the complete set of
 160 job attribute constraints associated with the containing printer object.

161 The attribute values in job-constraints-col may also be in range syntax, if the
 162 corresponding job attribute has integer syntax. This represents the included (or
 163 excluded, if the attribute is named in job-constraints-inverted) range of values for
 164 that attribute within that constraint.

165 1.4.2.2. job-constraints-inverted (1setOf type2 keyword)

166 The job-constraints-inverted attribute lists the names of other attributes in the
 167 current job-constraints-col value whose comparison logic must be inverted. That is,
 168 the values of named attributes are to be *excluded* ("not equal to" values) from the
 169 constraint. If an attribute name is not included in the job-constraints-inverted
 170 attribute, then that attribute's values are to be included ("equal to" values) in the
 171 constraint.

172 You can think of the each attribute in a job-constraints-cols value as AND-ed
 173 together to express a disallowed combination of options: "(attr1 == values) AND
 174 (attr2 == values) AND ...". The job constraints-inverted attribute lists those
 175 attribute/value comparisons which are to be "!=" instead of "==".

176 1.4.2.3. Example

177 Here is an example of how the job-constraints-col attribute can be used to express
 178 various printer constraints. The example is expressed in pseudo-code with curly
 179 brackets enclosing each collection value and attributes within each collection are
 180 shown on separate lines with commas separating the values (this is the PAPI text
 181 encoding format documented in Appendix A of the PAPI spec, with the additional
 182 of not-legal-syntax comments in "/* ... */" to help describe the examples):

```

183 job-constraints-col =
184
185     /* Constraint: no high print quality with 240 dpi resolution */
186     /* (print-quality == high) AND (printer-resolution == 240dpi) */
187     {
188         print-quality = high
189         printer-resolution = 240dpi
190     },
191
192     /* Constraint: no transparency with duplex */
193     /* (sides != one-sided) AND (media == transparency) */
194     {
195         job-constraints-inverted = sides
196         sides = one-sided
197         media = transparency
198     },
199
200     /* Constraint: no finishing with heavy-stock media */
201     /* (finishings != none) AND (media == heavy-stock) */
202     {
203         job-constraints-inverted = finishing
204         finishings = none
205         media = heavy-stock
206     },
207
208     /* Constraint: no duplex printing of A4 paper in landscape */
209     /* (sides != one-sided) AND (media == A4) AND */
210     /* (orientation-requested == landscape) */
211     {
212         job-constraints-inverted = sides
213         sides = one-sided
214         media = A4
215         orientation-requested = landscape
216     },
217
218     /* Constraint: no duplex printing of COM-10 envelopes */
219     /* (sides != one-sided) AND (media == envelope) AND */
220     /* (media-size == com10) */
221     {
222         job-constraints-inverted = sides
223         sides = one-sided
224         media = envelope
225         media-size = com10
226     },
227
228     /* Constraint: no stapling of greater than 50 sheets */
  
```

```
229      /* (finishings == staple) AND (job-media-sheets > 50)      */
230      {
231          job-constraints-inverted = job-media-sheets
232          finishings = staple
233          job-media-sheets = 1-50
234      }
235
236  };
237
```

238 **1.4.3. Validation Function**

239 The API used by the application to validate print job attributes against printer
240 capabilities is the `papiJobValidate` function. See the description of that function for
241 further details.

242 **Appendix A. Change History**

243 **Version 0.3 (December 6, 2002)**

244

- 245
- Clarified syntax of job-constraints-col example.
 - Added usage note clarifying the definition of "constraint" as used in this document.
 - Miscellaneous wording and typo corrections.

249

250 **Version 0.2 (November 21, 2002)**

251

- 252
- Added third capabilities usage to "Introduction".
 - Added paragraph about boolean logic under "Support returning constraints" objective. Also clarified wording of how this can be used to improve print dialogs.
 - Changed "Support returning composite features" to "Support logically grouping features" so that the objective does not imply a specific solution.
 - Removed "Support Device Object" objective.
 - Added "job-constraints-col" attribute.
 - Added "job-constraints-inverted" attribute.

261

262 **Version 0.1 (September 25, 2002)**

263

- 264
- Original draft version

265

266

267

268

269

270

<i>End of Document</i>
