

March 20, 2013
Technical Whitepaper



The Printer Working Group

Smith Kennedy 3/20/13 11:15 PM

Deleted: February 5

Smith Kennedy 3/20/13 11:15 PM

Deleted: Working Draft

IPP Client Use Best Practices

Status: Interim

Smith Kennedy 3/20/13 11:15 PM

Deleted: Initial

Abstract: This document enumerates a number of tasks that are commonly performed by a client in the process of interacting with a print service, and explores options for how the Internet Printing Protocol (IPP) may be used to perform those tasks, some of which are preferred and others that are less than optimal.

This document is a PWG Working Draft. For a definition of a "PWG Working Draft", see: <ftp://ftp.pwg.org/pub/pwg/general/pwg-process30.pdf>

This document is available electronically at:

<ftp://ftp.pwg.org/pub/pwg/general/templates/tb-ipp-best-practices-20130205.pdf>

1 Copyright © 2013 The Printer Working Group. All rights reserved.

2 This document may be copied and furnished to others, and derivative works that comment
3 on, or otherwise explain it or assist in its implementation may be prepared, copied,
4 published and distributed, in whole or in part, without restriction of any kind, provided that
5 the above copyright notice, this paragraph and the title of the Document as referenced
6 below are included on all such copies and derivative works. However, this document itself
7 may not be modified in any way, such as by removing the copyright notice or references to
8 the IEEE-ISTO and the Printer Working Group, a program of the IEEE-ISTO.

9 Title: *IPP Client Use Best Practices*

10 The IEEE-ISTO and the Printer Working Group DISCLAIM ANY AND ALL WARRANTIES,
11 WHETHER EXPRESS OR IMPLIED INCLUDING (WITHOUT LIMITATION) ANY IMPLIED
12 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

13 The Printer Working Group, a program of the IEEE-ISTO, reserves the right to make
14 changes to the document without further notice. The document may be updated, replaced
15 or made obsolete by other documents at any time.

16 The IEEE-ISTO takes no position regarding the validity or scope of any intellectual
17 property or other rights that might be claimed to pertain to the implementation or use of the
18 technology described in this document or the extent to which any license under such rights
19 might or might not be available; neither does it represent that it has made any effort to
20 identify any such rights.

21 The IEEE-ISTO invites any interested party to bring to its attention any copyrights, patents,
22 or patent applications, or other proprietary rights which may cover technology that may be
23 required to implement the contents of this document. The IEEE-ISTO and its programs
24 shall not be responsible for identifying patents for which a license may be required by a
25 document and/or IEEE-ISTO Industry Group Standard or for conducting inquiries into the
26 legal validity or scope of those patents that are brought to its attention. Inquiries may be
27 submitted to the IEEE-ISTO by e-mail at: ieee-isto@ieee.org.

28 The Printer Working Group acknowledges that the IEEE-ISTO (acting itself or through its
29 designees) is, and shall at all times, be the sole entity that may authorize the use of
30 certification marks, trademarks, or other special designations to indicate compliance with
31 these materials.

32 Use of this document is wholly voluntary. The existence of this document does not imply
33 that there are no other ways to produce, test, measure, purchase, market, or provide other
34 goods and services related to its scope.
35

Smith Kennedy 3/20/13 11:15 PM
Deleted: February 5

36 **About the IEEE-ISTO**

37 The IEEE-ISTO is a not-for-profit corporation offering industry groups an innovative and
38 flexible operational forum and support services. The IEEE-ISTO provides a forum not only
39 to develop standards, but also to facilitate activities that support the implementation and
40 acceptance of standards in the marketplace. The organization is affiliated with the IEEE
41 (<http://www.ieee.org/>) and the IEEE Standards Association (<http://standards.ieee.org/>).

42 For additional information regarding the IEEE-ISTO and its industry programs visit:

43 <http://www.ieee-isto.org>

Unknown
Field Code Changed
Unknown
Field Code Changed
Unknown
Field Code Changed

44 **About the IEEE-ISTO PWG**

45 The Printer Working Group (or PWG) is a Program of the IEEE Industry Standards and
46 Technology Organization (ISTO) with member organizations including printer
47 manufacturers, print server developers, operating system providers, network operating
48 systems providers, network connectivity vendors, and print management application
49 developers. The group is chartered to make printers and the applications and operating
50 systems supporting them work together better. All references to the PWG in this
51 document implicitly mean “The Printer Working Group, a Program of the IEEE ISTO.” In
52 order to meet this objective, the PWG will document the results of their work as open
53 standards that define print related protocols, interfaces, procedures and conventions.
54 Printer manufacturers and vendors of printer related software will benefit from the
55 interoperability provided by voluntary conformance to these standards.

56 In general, a PWG standard is a specification that is stable, well understood, and is
57 technically competent, has multiple, independent and interoperable implementations with
58 substantial operational experience, and enjoys significant public support.

59 For additional information regarding the Printer Working Group visit:

60 <http://www.pwg.org>

61 Contact information:

62 The Printer Working Group
63 c/o The IEEE Industry Standards and Technology Organization
64 445 Hoes Lane
65 Piscataway, NJ 08854
66 USA
67

68 **About the Internet Printing Protocol Work Group**

69 The Internet Printing Protocol (IPP) working group has developed a modern, full-featured
70 network printing protocol, which is now the industry standard. IPP allows a print client to
71 query a printer for its supported capabilities, features, and parameters to allow the
72 selection of an appropriate printer for each print job. IPP also provides job information prior
73 to, during, and at the end of job processing.

74 For additional information regarding IPP visit:

75 <http://www.pwg.org/ipp/>

76 Implementers of this specification are encouraged to join the IPP mailing list in order to
77 participate in any discussions of the specification. Suggested additions, changes, or
78 clarification to this specification, should be sent to the IPP mailing list for consideration.

79

Smith Kennedy 3/20/13 11:15 PM
Deleted: February 5

Table of Contents

80

81 1. Introduction 6

82 2. Terminology 6

83 2.1 Conformance Terminology 6

84 2.2 Other Terminology 6

85 2.3 Acronyms and Organizations 6

86 3. Requirements 7

87 3.1 Rationale 7

88 3.2 Use Cases 7

89 3.2.1 Developer Implementing New IPP Client Support 7

90 3.2.2 Developer Implementing New IPP Printer Support 7

91 3.3 Out of Scope 7

92 3.4 Design Requirements 7

93 4. Tasks and Implementation Alternatives 8

94 4.1 Create A Relationship With A Printer 8

95 4.1.1 Discover And Select A Printer Via A Discovery Protocol 8

96 4.1.2 Select A Printer Via User Provided DNS Hostname Or Raw Ipv4 / Ipv6 Address 9

97 4.2 Validate User Access to Printer 10

98 4.3 Get Printer Options 10

99 4.4 Check constraints between presented options 11

100 4.5 Submitting a Print Job 12

101 4.5.1 Submitting a print job with document data 12

102 4.5.2 Submitting a print job with document references 13

103 4.6 Monitoring print job status 14

104 4.7 Canceling a Print Job 15

105 4.8 Getting printer supplies status 16

106 5. Attributes and Their Use in Operations 16

107 5.1 Explicit "document-format" Selection 16

108 5.2 Prefer "media-col" Attribute To "media" Attribute 16

109 5.3 Prefer "finishings-col" Attribute To "finishings" Attribute 17

110 5.4 Using "ipp-attribute-fidelity" 17

111 5.5 Using "pdl-override" 17

112 6. HTTP Protocol Usage 17

113 6.1 HTTP/1.1 Expect Header 17

114 7. Security Considerations 18

115 8. References 18

116 8.1 Informative References 18

117 9. Authors' Addresses 19

118 10. Change History 19

119 10.1 February 5, 2013 19

120 10.2 March 20, 2013 20

121

122

123

124

Smith Kennedy 3/20/13 11:16 PM
Deleted: 1. Introduction . 6 [1]

Smith Kennedy 3/20/13 11:15 PM
Moved down [1]: 10.

Smith Kennedy 3/20/13 11:15 PM
Moved (insertion) [1]

128 **1. Introduction**

129 The use case descriptions below represent stages or sub-tasks that users perform in the
130 process of using a printer. Each of these below include a textual description as well as a
131 series of workflow options for how it might be implemented using IPP. Each workflow
132 option will be informally labeled according to its perceived quality, using the set of labels
133 {"BAD", "POOR", "GOOD", "BETTER", "BEST"}, that are ordered from least desirable to
134 most desirable.

135 **2. Terminology**

136 **2.1 Conformance Terminology**

137 Capitalized terms, such as MUST, MUST NOT, RECOMMENDED, REQUIRED, SHOULD,
138 SHOULD NOT, MAY, and OPTIONAL, have special meaning relating to conformance as
139 defined in Key words for use in RFCs to Indicate Requirement Levels [RFC2119]. The
140 term CONDITIONALLY REQUIRED is additionally defined for a conformance requirement
141 that applies to a particular capability or feature.

142 **2.2 Other Terminology**

143 *Capitalized Term In Italics*: definition of the term with any references as appropriate.

144 **2.3 Acronyms and Organizations**

145 *IANA*: Internet Assigned Numbers Authority, <http://www.iana.org/>

146 *IETF*: Internet Engineering Task Force, <http://www.ietf.org/>

147 *ISO*: International Organization for Standardization, <http://www.iso.org/>

148 *PWG*: Printer Working Group, <http://www.pwg.org/>

149

150

151 3. Requirements

152 3.1 Rationale

153 The Internet Printing Protocol/1.1: Implementor's Guide [RFC3196] was ratified in
154 November 2001. Since that time many extensions to IPP have been ratified, and the
155 scope of use of IPP has grown considerably. Given all these extensions to IPP,
156 implementers would benefit from an updated best practices document that covers the use
157 of these extensions, as well as the core of IPP that has remained unchanged, to assist
158 implementers in their efforts to deliver a quality client experience.

159 3.2 Use Cases

160 3.2.1 Developer Implementing New IPP Client Support

161 Garrett is a developer working on a new client platform that is adding system-level printing
162 support. Many printers support IPP Everywhere [PWG5100.14], so he plans to implement
163 printing support in his client platform using this standard as well. But IPP Everywhere and
164 its related standards don't describe how best to use IPP for the various tasks his software
165 must perform, in order to deliver a quality client user experience. He finds RFC 3196 but
166 its recommendations are insufficient. Using the IPP Use Best Practices document, he is
167 able to avoid some common design pitfalls and quickly deliver a quality IPP client
168 experience.

169 3.2.2 Developer Implementing New IPP Printer Support

170 Duncan is a firmware developer at a printer vendor creating a new printer that implements
171 IPP Everywhere. In reading the IPP Client Use Best Practices, he can more easily
172 anticipate how some segment of clients implemented according to these practices are
173 likely to behave, and more rapidly understand how the various operations can be used with
174 one another to achieve certain tasks.

175 3.3 Out of Scope

176 The following are considered out of scope for this specification:

- 177 1. Specifications to extend or replace portions of the Internet Printing Protocol itself
- 178 2. Normative requirements regarding user experience

179 3.4 Design Requirements

180 The design requirements for this specification are:

- 181 1. Explore tasks performed by client implementations
182 2. Enumerate a series of alternatives
183 3. Rank those options according to a non-numeric qualitative grading scheme

184 4. Tasks and Implementation Alternatives

185 For a number of tasks, the set of IPP operations provides a rich enough set of semantics
186 that it is possible to perform those tasks in a few different ways. In this section a number
187 of common tasks will be enumerated, and some alternatives for how those tasks might be
188 performed will be evaluated.

189 4.1 Create A Relationship With A Printer

190 You can't print to a printer if you cannot establish a connection to it. Historically,
191 connecting to a printer to establish a "relationship" with it meant identifying a printer and
192 then creating a persistent local records and resources for that printer relationship with your
193 system's print spooler. This was called a "print queue", and it involved binding drivers to
194 create the relationships needed to communicate at the different levels, and then keeping
195 record of that set of relationships so that it could be re-used at a later time. The set of
196 printers or other devices the user's system might encounter was relatively small and fairly
197 static.

198 More recent re-thinking of this relationship between client and printer has resulted in more
199 "dynamic" relationship creation, where universal drivers can interrogate a device hosting a
200 print service using a standardized protocol solution stack, and using that dynamically
201 ascertain and update print service attributes. In this paradigm, a "persistent" print service
202 record is more like a Web browser bookmark.

203 Both paradigms still require a method of identifying the target devices. That can be done
204 using dynamic service discovery protocols where the services respond to discovery
205 requests, or explicitly by name (host name or raw IPv4/IPv6 address).

206 4.1.1 Discover And Select A Printer Via A Discovery Protocol

207 Discovery protocols are used to identify instances of print services or printers by searching
208 the network for service types or device types. This helps the user by making it so that they
209 don't need to do a physical survey of devices' addresses.

210 Regardless of the actual discovery protocol used, the APIs driving the protocols generally
211 can be used in either a synchronous or asynchronous fashion. Unfortunately, many legacy
212 software systems (as well as developers) are accustomed to the synchronous model,
213 which is easily identified by the presence of a "refresh button". The synchronous model is
214 not as user friendly as the asynchronous model, but it is somewhat easier to write
215 programs in a synchronous way than an asynchronous way.

216 Options

- 217 • POOR:
 - 218 ○ Perform network discovery with a synchronous API
 - 219 ▪ Show progress bar
 - 220 ▪ Discovery.Start()
 - 221 ▪ sleep(X) where X is some reasonably short number of seconds
 - 222 ▪ Discovery.Stop()
 - 223 ○ Present the results of the discovery process
 - 224 ○ "Refresh" button restarts the process
 - 225 ▪ Why this is bad:
 - 226 • List contents can be stale
 - 227 • Results are not "live"
 - 228 • "Reset" button is unnecessary and is a crutch
 - 229 ○ User selects a printer and presses "Continue" or equivalent
- 230 • BETTER:
 - 231 ○ Perform network discovery with an asynchronous API
 - 232 ▪ Show List UI widget
 - 233 ▪ Discovery.Start() with a callback
 - 234 ▪ Callback is called when discovery responses (add or remove) are received
 - 235 ○ User selects a printer and presses "Continue" or equivalent
 - 236 ▪ Discovery.Stop()

238 4.1.2 Select A Printer Via User Provided DNS Hostname Or Raw Ipv4 / Ipv6 Address

239 In some cases a discovery protocol is either not adequate or unnecessary. Examples of
240 when this use case is encountered include pre-published names or addresses, and also
241 situations where the target device is not on the local link. (DNS-SD and WS-Discovery are
242 generally used for link-local discovery, though wide-area variants as well as LDAP systems
243 may also be used, but are frequently not for various reasons.)

244 For each of these options below, the assumption is that the client has been given an
245 address string, and should attempt to connect to the host at that address.

246 Options

- 247 • BAD:
 - 248 ○ Let each printer model make up its own path, and depend on some other
249 protocol to get the resource path
 - 250 ▪ IPP has no defined standard mechanism to enumerate the Printer
251 objects' resource paths
- 252 • POOR:
 - 253 ○ IPP Get-Printer-Attributes with printer-uri set to a URI that was manually
254 entered by the user

Smith Kennedy 3/20/13 11:15 PM
Deleted: February 5

Smith Kennedy 3/20/13 11:15 PM
Formatted: Font:Arial

Smith Kennedy 3/20/13 11:15 PM
Formatted ... [2]

Smith Kennedy 3/20/13 11:15 PM
Formatted ... [3]

Smith Kennedy 3/20/13 11:15 PM
Formatted ... [4]

Smith Kennedy 3/20/13 11:15 PM
Formatted ... [5]

Smith Kennedy 3/20/13 11:15 PM
Formatted ... [6]

Smith Kennedy 3/20/13 11:15 PM
Deleted: list

Smith Kennedy 3/20/13 11:15 PM
Formatted: Font:Arial

Smith Kennedy 3/20/13 11:15 PM
Deleted: The results

Smith Kennedy 3/20/13 11:15 PM
Formatted: Font:Arial

Smith Kennedy 3/20/13 11:15 PM
Formatted ... [7]

Smith Kennedy 3/20/13 11:15 PM
Formatted ... [8]

Smith Kennedy 3/20/13 11:15 PM
Formatted ... [9]

Smith Kennedy 3/20/13 11:15 PM
Deleted: callback

Smith Kennedy 3/20/13 11:15 PM
Formatted ... [10]

Smith Kennedy 3/20/13 11:15 PM
Formatted ... [11]

- 261 | ▪ [The](#) "ipp" URI scheme could be used to encode the hostname and the
 262 | resource path
 263 | ▪ Having the user enter the URI exposes too many details to the user,
 264 | including the detail about the fact that IPP is actually [being](#) used.
 265 | Users need not be aware of which print protocol is being used.
- 266 | • GOOD:
 - 267 | ○ IPP Get-Printer-Attributes with printer-uri set to a well-known Printer resource
 268 | path
 - 269 | ▪ "/ipp/print"
 - 270 | • BETTER:
 - 271 | ○ IPP Get-Printer-Attributes with printer-uri set to "/"
 - 272 | ○ [Examine](#) the "printer-uri-supported" attribute; use the first URI in the list
 - 273 | ○ IPP Get-Printer-Attributes with printer-uri set to first URI
 - 274 | • BEST:
 - 275 | ○ IPP Get-Services operation
 - 276 | ▪ Coming with System Control Service
 - 277 | ▪ Is this really going to be better?
 - 278 | • Yes, expected to have metadata associated with each URI
 - 279 | specifying the class of service

Smith Kennedy 3/20/13 11:15 PM
 Deleted: February 5

Smith Kennedy 3/20/13 11:15 PM
 Deleted: being

Smith Kennedy 3/20/13 11:15 PM
 Deleted: examine

280 4.2 Validate User Access to Printer

281 | Selecting a printer is misleading to the user if the user isn't allowed to use the selected
 282 | printer. Therefore, access restrictions should be validated before selection confirmation
 283 | (queue creation, etc.) is done on the client system.

284 Options

- 285 | • BAD:
 - 286 | ○ Do Nothing
 - 287 | ▪ The user may choose a printer but not be able to use it due to not
 288 | having access credentials (username or password or whatever) to use
 289 | that printer
- 290 | • GOOD:
 - 291 | ○ IPP Validate-Job operation
 - 292 | ▪ Use the defaults, but provide the credentials to allow the user access
 293 | to be determined

294 4.3 Get Printer Options

295 | Once [the user has selected](#) a printer, it is necessary for the print system to understand the
 296 | capabilities that the printer device's print service provides. This includes what print job
 297 | payload formats can be consumed by the print service, the available options and default
 298 | choices, and so forth. It also includes other information about the device itself, such as its

Smith Kennedy 3/20/13 11:15 PM
 Deleted: has been identified

Smith Kennedy 3/20/13 11:15 PM
Deleted: February 5

302 location. Some of this is done at relationship creation time (queue creation time), perhaps
303 by consulting information stored statically in the printer. It may be that this information can
304 all be retrieved from the printer itself. This is basically the print dialog's activity between
305 the time that the user performs an action to request that the print dialog be presented, and
306 the time that the dialog is presented to the user, populated with the available option
307 choices.

308 Options

- 309 • SAD:
 - 310 ○ Depend on a-priori knowledge about a particular model as a way of listing
 - 311 options for the model of device identified as the target
 - 312 ▪ Model specific print drivers fall in this bucket
- 313 • GOOD:
 - 314 ○ IPP Get-Printer-Attributes Operation
 - 315 ▪ Request includes no printer attributes; only operation attributes
 - 316 ▪ Reply will contain the job template attributes for all PDLs
 - 317 ○ Client guesses at what attributes may work or not work for a given PDL, or
 - 318 uses a-priori knowledge
- 319 • BETTER:
 - 320 ○ IPP Get-Printer-Attributes Operation
 - 321 ▪ Any specific attributes?
 - 322 ○ Process results; decide on a PDL
 - 323 ○ IPP Get-Printer-Attributes Operation
 - 324 ▪ Request includes the document-format attribute with value specifying
 - 325 the chosen PDL
 - 326 ▪ Reply will contain the job template attributes appropriately filtered
 - 327 ("colored") for that particular document-format

Smith Kennedy 3/20/13 11:15 PM
Deleted: request

Smith Kennedy 3/20/13 11:15 PM
Deleted: reply

Smith Kennedy 3/20/13 11:15 PM
Deleted: any

Smith Kennedy 3/20/13 11:15 PM
Deleted: request

Smith Kennedy 3/20/13 11:15 PM
Deleted: reply

328 4.4 Check constraints between presented options

329 Printer features and options are presented typically in a print dialog. Some of these have
330 states that have relationships with other options' states, where one cannot be in a
331 particular state if another one is too. These are known as constraints, and they must be
332 calculated any time the state of a control changes state. There are various ways that this
333 can be done.

334 Options

- 335 • POOR:
 - 336 ○ IPP Validate-Job
 - 337 ▪ Every time a control is changed, the client sends IPP Validate-Job
 - 338 with attribute values corresponding to current state of controls
- 339 • GOOD:

- 345 ○ IPP Validate-Job
- 346 ▪ When "Print" button is pressed, confirms the job creation / submission
- 347 will succeed (authentication, etc.)
- 348 ▪ Client depends on this operation to perform constraints validation
- 349 printer-side
- 350 • BETTER:
- 351 ○ IPP Get-Printer-Attributes
- 352 ▪ Printer Object implements job-constraints-supported
- 353 ▪ Printer Object implements job-resolvers-supported
- 354 ○ <Local processing of constraints>
- 355 ○ IPP Validate-Job
- 356 ▪ When "Print" button is pressed, confirms the job creation / submission
- 357 will succeed (authentication, etc.)
- 358 ▪ Constraints validation already handled client-side

Smith Kennedy 3/20/13 11:15 PM
Deleted: February 5

Smith Kennedy 3/20/13 11:15 PM
Deleted: when

Smith Kennedy 3/20/13 11:15 PM
Deleted: client

Smith Kennedy 3/20/13 11:15 PM
Deleted: device

Smith Kennedy 3/20/13 11:15 PM
Deleted: constraints

Smith Kennedy 3/20/13 11:15 PM
Deleted: device

Smith Kennedy 3/20/13 11:15 PM
Deleted: when

Smith Kennedy 3/20/13 11:15 PM
Deleted: constraints

359 4.5 Submitting a Print Job

360 Once the user has decided on options, the print job is generated and ultimately made
361 available to the printer in some fashion. There are several different ways that this may
362 occur.

363 4.5.1 Submitting a print job with document data

364 This is the classical way that a print job is sent from the client to the print service: first a job
365 is created, and then the job information and payload content are sent from the client to the
366 print service.

Smith Kennedy 3/20/13 11:15 PM
Deleted: "by value"

Smith Kennedy 3/20/13 11:15 PM
Deleted: ;

Smith Kennedy 3/20/13 11:15 PM
Deleted: by

367 Options

- 368 • POOR:
- 369 ○ IPP Print-Job
- 370 ▪ No pre-flight checks
- 371 ▪ The printer may reject it but only after it has been transmitted,
- 372 ▪ Better to check ticket and content types first.
- 373 • GOOD:
- 374 ○ IPP Validate-Job
- 375 ▪ Pre-flight checks the job by validating the job attributes, document
- 376 type, authentication and transport encryption upgrades (if needed)
- 377 ○ IPP Print-Job
- 378 ▪ Creates the job and sends the payload in one operation
- 379 ▪ However, the Job object's URI isn't usually known until the job
- 380 transmission is complete
- 381 ▪ Doesn't work well with flow-controlled (low-end) printers
- 382 • BETTER:
- 383 ○ IPP Validate-Job

Smith Kennedy 3/20/13 11:15 PM
Deleted: no

Smith Kennedy 3/20/13 8:33 PM
Comment [1]: IPP would never respond early; HTTP could respond early but that would be effectively a transport

Smith Kennedy 3/20/13 11:15 PM
Deleted: the

Smith Kennedy 3/20/13 11:15 PM
Deleted: in whole or in part

Smith Kennedy 3/20/13 11:15 PM
Deleted: better

Smith Kennedy 3/20/13 11:15 PM
Deleted: pre

Smith Kennedy 3/20/13 11:15 PM
Deleted: and

Smith Kennedy 3/20/13 11:15 PM
Deleted: creates

Smith Kennedy 3/20/13 11:15 PM
Deleted: however

Smith Kennedy 3/20/13 11:15 PM
Deleted: doesn't

- 403 ▪ Pre-flight checks the job by validating the job attributes and document
- 404 type, authentication and transport encryption upgrades (if needed)
- 405 ○ IPP Create-Job
- 406 ▪ Returns immediately with the job URI for monitoring and ticket
- 407 processing status
- 408 ▪ If there is a problem then Create-Job may fail the same as Validate-
- 409 Job would, but may not, which is why we do a Validate-Job first (so
- 410 that there isn't a zombie job there)
- 411 ▪ Once the job is created, the client will receive a list of the actual job
- 412 processing attributes from the IPP Printer.
- 413 ▪ Response to this operation will include xxx-actual job attributes that
- 414 could be used to detect substitutions that would be used by the Printer
- 415 Object. Observing this, the client may decide whether to cancel the
- 416 job rather than submit the document with this job. If the original job
- 417 was cancelled, the client could create another job with a new set of
- 418 attributes submitted, or error out and not submit a job at all
- 419 ▪ Allows an opportunity to perform a Cancel-Job operation during
- 420 document submission
- 421 ○ IPP Send-Document
- 422 ▪ Payload transmission is de-coupled from the creation of the job
- 423 ▪ Multiple documents can be sent to build up a compound job
- 424 ▪ Client MUST check to see if value of "multiple-document-jobs-
- 425 supported" is "true", to see if it is OK to do multiple Send-Document
- 426 operations to the same Job object.

4.5.2 Submitting a print job with document references

This is a slightly different way that a print job is sent from the client to the print service: a job is created and made available for retrieval by the print service, and when the print job the job information and job payload content are sent by the client to the print service.

Options

- 432 • POOR:
- 433 ○ IPP Print-URI
- 434 ▪ No pre-flight checks
- 435 ▪ Printer may reject it but only after it has been transmitted,
- 436 ▪ Better to check ticket and content types first
- 437 • GOOD:
- 438 ○ IPP Validate-Job
- 439 ▪ Pre-flight checks the job by validating the job attributes and document
- 440 type
- 441 ○ IPP Print-URI
- 442 ▪ Creates the job and sends a URL to where the payload can be
- 443 retrieved in one operation

Smith Kennedy 3/20/13 11:15 PM
Deleted: February 5

Smith Kennedy 3/20/13 11:15 PM
Deleted: pre

Smith Kennedy 3/20/13 11:15 PM
Deleted: returns

Smith Kennedy 3/20/13 11:15 PM
Deleted: if

Smith Kennedy 3/20/13 11:15 PM
Deleted: payload

Smith Kennedy 3/20/13 11:15 PM
Deleted: multiple

Smith Kennedy 3/20/13 11:15 PM
Deleted: <#>means that the client doesn't have to be prepared for an early HTTP response . . . [12]

Smith Kennedy 3/20/13 11:15 PM
Deleted: "by reference"

Smith Kennedy 3/20/13 11:15 PM
Deleted: no

Smith Kennedy 3/20/13 11:15 PM
Deleted: the printer

Smith Kennedy 3/20/13 11:15 PM
Deleted: in whole or in part.

Smith Kennedy 3/20/13 11:15 PM
Deleted: better

Smith Kennedy 3/20/13 11:15 PM
Deleted: .

Smith Kennedy 3/20/13 11:15 PM
Deleted: pre

Smith Kennedy 3/20/13 11:15 PM
Deleted: creates

- 461 ▪ Printer Object "pulls" the document file rather than being given it by
- 462 the client
- 463 ▪ However, the Job object's URI isn't usually known until the job
- 464 transmission is complete
- 465 ▪ Printer may respond with client-error-document-access-error status
- 466 code, or might add document-access-error to job-state-reasons
- 467 ▪ URI may not be accessible at time of processing
- 468 • BETTER:
 - 469 o IPP Validate-Job
 - 470 ▪ Pre-flight checks the job by validating the job attributes and document
 - 471 type
 - 472 o IPP Create-Job
 - 473 ▪ Returns immediately with the job URI for monitoring and ticket
 - 474 processing status
 - 475 ▪ If there is a problem then Create-Job will fail the same as Validate-Job
 - 476 would
 - 477 o IPP Send-URI
 - 478 ▪ Payload URI transmission is de-coupled from the creation of the job
 - 479 ▪ Printer may respond with client-error-document-access-error status
 - 480 code, or might add document-access-error to job-state-reasons
 - 481 ▪ URI may not be accessible at time of processing
 - 482 • (How to handle this appropriately? What recommendations
 - 483 should be provided?)

4.6 Monitoring print job status

485 While the print job is being processed, users may wish to know whether it is proceeding
486 successfully, or whether there are conditions that they need to handle that are preventing
487 processing from proceeding, such as a media jam, open covers, marking agents depleted,
488 and so forth.

489 For those options below that involve polling the Printer Object, the degree to which the
490 option is better or worse is due in no small part to the polling frequency. The interval
491 should be tuned so that the frequency of queries is not so great that it burdens the Printer
492 Object or Job Object or the network, but not so small that there is an undesirable lag
493 between when an event occurs and when the user is notified. It is certainly NOT a best
494 practice in any case if a client is polling as fast as the network can handle traffic.

Options

- 495 • POOR:
 - 496 o IPP Get-Jobs / IPP Get-Printer-Attributes
 - 497 ▪ Monitor the value of the printer-state attribute and the state of all jobs
 - 498 ▪ Not precise; polling for status without knowing the actual job ID
 - 499 ▪ Polling is generally not desirable

Smith Kennedy 3/20/13 11:15 PM
Deleted: February 5

Smith Kennedy 3/20/13 11:15 PM
Deleted: <#>print service retrieves the payload file itself - ... [13]

Smith Kennedy 3/20/13 11:15 PM
Deleted: <#>doesn't work well with flow-controlled (low-end) printers -

Smith Kennedy 3/20/13 11:15 PM
Deleted: pre

Smith Kennedy 3/20/13 11:15 PM
Deleted: returns

Smith Kennedy 3/20/13 11:15 PM
Deleted: if

Smith Kennedy 3/20/13 11:15 PM
Deleted: payload

Smith Kennedy 3/20/13 11:15 PM
Deleted: <#>means that the client doesn't have to be prepared for an early HTTP response - ... [14]

Smith Kennedy 3/20/13 11:15 PM
Formatted

Smith Kennedy 3/20/13 11:15 PM
Deleted: learned before job payload is sent

Smith Kennedy 3/20/13 8:33 PM
Comment [2]: Should we put specific recommendations as to a polling frequency here?

Smith Kennedy 3/20/13 11:15 PM
Deleted: monitor

Smith Kennedy 3/20/13 11:15 PM
Deleted: <#>polling (lame) -

- 518 [See above regarding polling intervals](#)
- 519 • GOOD:
 - 520 ○ IPP Get-Job-Attributes / IPP Get-Printer-Attributes
 - 521 ▪ [Monitor the value of printer state attribute as well as targeted monitoring of a specific job's status](#)
 - 522 ▪ [Polling is generally not desirable](#)
 - 523 ▪ [See above regarding polling intervals](#)
- 524 • BETTER:
 - 525 ○ IPP Create-Printer-Subscriptions / IPP Get-Notifications / IPP Get-Job-Attributes
 - 526 ▪ [Asynchronous / long running queries for notifications that don't require polling](#)
 - 527 ▪ [When you see that a job has completed, query the state of that job at that time](#)
 - 528 ▪ [Printer state changes will be provided by subscribing to the printer; subscribing to the job will provide less information and not be as useful](#)

535 4.7 Canceling a Print Job

536 It may be that the user wants to terminate a job before it has been fully processed, for
537 whatever reason. There are things that must be done to ensure that the client has
538 decisively cleaned up the state of the Job Object if the client is responsible for canceling
539 the job. Clients' leaving broken Job objects on the Print service is bad behavior.

540 There is also a dependency between the options below and how the job was submitted.

541 Options

- 542 • BAD:
 - 543 ○ [IPP Print-Job operation](#)
 - 544 ○ [Client stops sending chunks](#)
- 545 • POOR:
 - 546 ○ [IPP Print-Job operation](#)
 - 547 ○ Client stops sending chunks
 - 548 ○ IPP Cancel-Job operation request for the job via a second connection, which
549 for some printers could result in a PDL interpreter hang because the last
550 chunk sent didn't stop on a "statement" boundary
- 551 • GOOD:
 - 552 ○ [IPP Create-Job operation](#)
 - 553 ○ [IPP Send-Document operation](#)
 - 554 ▪ [Potentially truncating job during Send-Document payload transmission](#)
 - 555 ○ IPP Cancel-Job operation

Smith Kennedy 3/20/13 11:15 PM
Deleted: February 5

Smith Kennedy 3/20/13 11:15 PM
Deleted: targets

Smith Kennedy 3/20/13 11:15 PM
Deleted: specific job status - ... [15]

Smith Kennedy 3/20/13 11:15 PM
Deleted: issues (top cover open, etc.) that block a job

Smith Kennedy 3/20/13 11:15 PM
Deleted:)

Smith Kennedy 3/20/13 11:15 PM
Formatted

Smith Kennedy 3/20/13 11:15 PM
Deleted: (lame)

Smith Kennedy 3/20/13 11:15 PM
Deleted: Job

Smith Kennedy 3/20/13 11:15 PM
Deleted: asynchronous

Smith Kennedy 3/20/13 11:15 PM
Deleted: If Print-Job is used, but the document payload is not completely transmitted, then is a Job object even created? (Is this true in all cases? It is - provide a cross-reference.) Also, if Create-Job / Send-Document is used and the Cancel-Job is sent during the Send-Document operation submission, then the job object would still need to be cleaned up by the client that created the Job Object. (Is this true? Provide a cross reference.)

Smith Kennedy 3/20/13 11:15 PM
Deleted: <#>Client stop sending data and close the connection - ... [16]

Smith Kennedy 3/20/13 11:15 PM
Deleted: but doesn't close the connection because it is expecting an IPP operation reply.

Smith Kennedy 3/20/13 11:15 PM
Deleted: <#>But if the client stops with a zero length chunk then the IPP stack will know that transport is comp ... [17]

586 4.8 Getting printer supplies status

587 Some administrative tasks, like checking consumables levels, are presented to end users
588 in some cases, such as during print job status or in print dialogs. This is useful to end-
589 users and should be supported.

590 Options

- 591 • POOR:
 - 592 ○ Don't use IPP but use some proprietary protocol or platform-specific
 - 593 extension to IPP
 - 594 ▪ The point is to use only IPP extensions based on open standards (i.e.
 - 595 PWG standard) and this violates that core principle
- 596 • GOOD:
 - 597 ○ IPP Get-Printer-Attributes
 - 598 ▪ Printer must implement JPS3 "printer-supply" attribute
- 599 • BETTER
 - 600 ○ IPP Create-Printer-Subscription operation + IPP Get-Notifications operation
 - 601 ○ IPP Get-Printer-Attributes operations

602 603 5. Attributes and Their Use in Operations

604 Some attributes that IPP has labeled as optional should always be used as a best practice.
605 Below are some of these attributes and how they should be used in various contexts.

606 5.1 Explicit "document-format" Selection

607 While IPP Printer Objects provide a default document format (which is known via the
608 document-format-default attribute), as a general principle, it is much better for a client to
609 explicitly provide the document-format attribute with all operations relating to validating or
610 submitting a document payload to the printer (Validate-Job, Print-Job, Send-Document).

611 5.2 Prefer "media-col" Attribute To "media" Attribute

612 Given a Printer Object that supports both "media" and "media-col" attributes, a client
613 should prefer to include the "media-col" attribute with operations that accept one of these
614 attributes. This is true for when "media" and "media-col" are top-level attributes as well as
615 when "media" or "media-col" may be included within other collection attributes, such as
616 "job-sheets", "job-error-sheet", "job-accounting-sheets", and others.

Smith Kennedy 3/20/13 11:15 PM
Deleted: February 5

Smith Kennedy 3/20/13 11:15 PM
Moved down [2]: <#>BETTER

Smith Kennedy 3/20/13 11:15 PM
Deleted: <#>Client stops sending chunks and closes the connection .

Smith Kennedy 3/20/13 11:15 PM
Deleted: <#>: [18]

Smith Kennedy 3/20/13 11:15 PM
Deleted:

Smith Kennedy 3/20/13 11:15 PM
Deleted: printer

Smith Kennedy 3/20/13 11:15 PM
Moved (insertion) [2]

Smith Kennedy 3/20/13 11:15 PM
Deleted: [19]

626 **5.3 Prefer "finishings-col" Attribute To "finishings" Attribute**

627 Given a Printer Object that supports both "finishings" and "finishings-col" attributes, a client
628 should prefer to include the "finishings-col" attribute with operations that accept one of
629 these attributes.

630 **5.4 Using "ipp-attribute-fidelity"**

631 TBD

632 **5.5 Using "pdl-override"**

633 TBD

634 **6. HTTP Protocol Usage**

635 IPP currently uses HTTP/1.1 for its transport. IPP/2.0 and other IPP specifications have
636 specified some of the facilities of HTTP that IPP clients and servers should support in
637 order to provide the semantics that IPP needs to provide a great user experience. Even
638 so, there are best practices that should be followed.

639 **6.1 HTTP/1.1 Expect Header**

640 As defined in [RFC 2616 "HTTP/1.1"], the "Expect" header allows the client to check with
641 the server on the HTTP connection negotiation before sending the HTTP request payload.

642 The IPP client should implement the following:

- 643 • On first request to a printer, include the "Expect: 100-continue" header.
- 644 • Wait up to 1 second for a response.
- 645 • If no response is received, remember this for the next request so that you don't
646 have the 1-second delay; continue sending the request.
- 647 • If a 100 (continue) status code is returned, continue sending the request
- 648 • If a 301 (moved permanently) or 302 (moved temporarily) status code is returned,
649 redirect the request to the new URI *or* fail/report an error depending on the
650 security requirements of the Client (redirection is generally unexpected)
- 651 • If a 400 (Bad Request) status code is returned, remember this (don't use Expect
652 header) and re-send the POST request. This Printer is technically non-conforming
653 since it fails RFC 2616 requirements for a HTTP/1.1 server.

Smith Kennedy 3/20/13 11:15 PM

Deleted: February 5

- 654
- 655
- If a 401 status code is returned, re-send the POST request with the requested credentials.
- 656
- If a 403 status code is returned, fail/report an error.
- 657
- If a 426 status code is returned, send an OPTIONS * request to upgrade to TLS, then re-send the POST request.
- 658
- 659

660 The IPP server should implement the following:

- 661
- Return status code 403 for unauthorized client addresses when the HTTP level authentication or authorization is not adequate
- 662
- Return status code 200 with an IPP response containing the client-error-not-authorized status code when the IPP level authentication or authorization is not adequate
- 663
- Status codes 301 and 302 are not recommended
- 664
- Return status code 400 only if problems are detected with the HTTP request itself
- 665
- Return status code 200 with an IPP response containing the client-error-bad-request status code if problems are detected with the IPP operation
- 666
- 667
- 668
- 669

670 7. Security Considerations

671 TBD,

- 672
- What you might do to ensure that the documents submitted remain private
- 673
- Using the [IPPS URI]

Smith Kennedy 3/20/13 11:15 PM

Deleted: .

Smith Kennedy 3/20/13 11:15 PM

Deleted: <#>IANA Considerations ... [20]

674 8. References

675 8.1 Informative References

- 676
- 677
- 678
- [PWG5100.12] R. Bergman, H. Lewis, I. McDonald, M. Sweet, "IPP/2.0 Second Edition", PWG 5100.12-2011, February 2011, <ftp://ftp.pwg.org/pub/pwg/candidates/cs-ipp20-20110214-5100.12.pdf>
- 679
- 680
- 681
- [PWG5100.14] F. Last author list or standards body, "IPP Everywhere", 5100.14-2013, January 2013, <ftp://ftp.pwg.org/pub/pwg/candidates/cs-ippeve10-20130128-5100.14.pdf>
- 682
- 683
- 684
- [RFC2616] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, <http://www.ietf.org/rfc/rfc2616.txt>

Smith Kennedy 3/20/13 11:15 PM

Deleted: <#>Normative References ... [21]

Smith Kennedy 3/20/13 11:15 PM

Deleted: February 5

- 690 [RFC2910] R. Herriot, S. Butler, P. Moore, R. Tuner, J. Wenn, "Internet Printing
691 Protocol/1.1: Encoding and Transport", RFC 2910, September 2000,
692 <http://www.ietf.org/rfc/rfc2910.txt>
- 693 [RFC2911] R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet
694 Printing Protocol/1.1: Model and Semantics", RFC 2911, September
695 2000, <http://www.ietf.org/rfc/rfc2911.txt>
- 696 [RFC3196] T. Hastings, C. Manros, P. Zehler, C. Kugler, H. Holst, "Internet
697 Printing Protocol/1.1: Implementer's Guide", RFC 3196, November
698 2001, <http://www.ietf.org/rfc/rfc3196.txt>
- 699 [RFC6762] S. Cheshire, M. Krochmal, "Multicast DNS", RFC 6762, February
700 2013, <http://www.ietf.org/rfc/rfc6762.txt>
- 701 [RFC6763] S. Cheshire, M. Krochmal, "DNS-Based Service Discovery", RFC
702 6763, February 2013, <http://www.ietf.org/rfc/rfc6763.txt>
- 703 [REFERENCE] F. Last author list or standards body, "Title of referenced document",
704 Document Number, Month YYYY, URL (if any)
- 705

706 9. Authors' Addresses

707 Primary authors:

708 Smith Kennedy
709 Hewlett-Packard Co.
710 11311 Chinden Blvd. MS 506
711 Boise, ID 83714
712 smith.kennedy@hp.com

Smith Kennedy 3/20/13 11:15 PM

Deleted: 83702

713 The authors would also like to thank the following individuals for their contributions to this
714 [white paper](#):

715 Evan Williams - Kentucky State Board of Recreation

Smith Kennedy 3/20/13 11:15 PM

Deleted: standard

716 10. Change History

717 10.1 February 5, 2013

718 Initial revision.

Smith Kennedy 3/20/13 11:15 PM
Deleted: February 5

721 **10.2 March 20, 2013**

722 Resolved issues from feedback provided during the IPP conference call on February 25,
723 2013, as documented in teleconference meeting minutes and author's own notes.

724 1. Added Validate-Job operation as operation to be used during printer selection
725 process to validate access by client / user

726 2. Replaced previous Section 5 "Conformance Requirements" with new Section 5
727 "Attributes and Their Use in Operations"

728 3. Replaced previous Section 6 "Internationalization Considerations" with new Section
729 6 "HTTP Protocol Usage"

730 4. Added updated list of references