

1 INTERNET-DRAFT -- BakeOff2 issue 1-36 resolutions are flagged as **Issue nn** and can be searched for
2 draft-ietf-ipp-model-v11-~~0204~~.txt

3 R. deBry

4 ~~IBM Corporation~~ Utah Valley State College

5 T. Hastings (editor)

6 Xerox Corporation

7 R. Herriot

8 Xerox Corporation

9 S. Isaacson

10 Novell, Inc.

11 P. Powell

12 Astart Technologies

13 ~~February 17~~ May 10 1998

14
15 Internet Printing Protocol/1.1: Model and Semantics

16 Copyright (C) The Internet Society (1999). All Rights Reserved.

17 Status of this Memo

18 This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of
19 [RFC2026]. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its
20 areas, and its working groups. Note that other groups may also distribute working documents as
21 Internet-Drafts.

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

25 The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>

26 The list of Internet-Draft Shadow Directories can be accessed as <http://www.ietf.org/shadow.html>.

27 Abstract

28 This document is one of a set of documents, which together describe all aspects of a new Internet
29 Printing Protocol (IPP). IPP is an application level protocol that can be used for distributed printing
30 using Internet tools and technologies. This document describes a simplified model consisting of abstract
31 objects, their attributes, and their operations that is independent of encoding and transport. The model
32 consists of a Printer and a Job object. A Job optionally supports multiple documents. IPP 1.1 semantics
33 allow end-users and operators to query printer capabilities, submit print jobs, inquire about the status of
34 print jobs and printers, cancel, hold, release, and restart print jobs. IPP 1.1 semantics allow operators to
35 pause, resume, and purge (jobs from) Printer objects. This document also addresses security,
36 internationalization, and directory issues.

37 The full set of IPP documents includes:

38 Design Goals for an Internet Printing Protocol [~~RFC2567~~ IPP-REQ]

deBry, Hastings, Herriot, Isaacson, Powell

[Page 1]

Expires November 10, 1999

39 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [[RFC2568](#)~~IPP-~~
40 ~~RAT~~]
41 Internet Printing Protocol/1.1: Model and Semantics (this document)
42 Internet Printing Protocol/1.1: Encoding and Transport [IPP-PRO]
43 Internet Printing Protocol/1.1: Implementer's Guide [IPP-IIG]
44 Mapping between LPD and IPP Protocols [[RFC2569](#)~~IPP-LPD~~]
45

46 The "Design Goals for an Internet Printing Protocol" document takes a broad look at distributed printing
47 functionality, and it enumerates real-life scenarios that help to clarify the features that need to be
48 included in a printing protocol for the Internet. It identifies requirements for three types of users: end
49 users, operators, and administrators. It calls out a subset of end user requirements that are satisfied in
50 IPP/1.0. Operator and administrator requirements are out of scope for version 1.0. A few OPTIONAL
51 operator operations have been added to IPP/1.1.

52 The "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol" document
53 describes IPP from a high level view, defines a roadmap for the various documents that form the suite of
54 IPP specifications, and gives background and rationale for the IETF working group's major decisions.

55 The "Internet Printing Protocol/1.1: Encoding and Transport" document is a formal mapping of the
56 abstract operations and attributes defined in the model document onto HTTP/1.1. It defines the encoding
57 rules for a new Internet MIME media type called "application/ipp". This document also defines the rules
58 for transporting over HTTP a message body whose Content-Type is "application/ipp". This document
59 defines a new scheme named 'ipp' for identifying IPP printers and jobs. Finally, this document defines
60 [interoperability](#) rules for supporting IPP/1.0 clients. **Issue 33**

61 The "Internet Printing Protocol/1.1: Implementer's Guide" document gives insight and advice to
62 implementers of IPP clients and IPP objects. It is intended to help them understand IPP/1.1 and some of
63 the considerations that may assist them in the design of their client and/or IPP object implementations.
64 For example, a typical order of processing requests is given, including error checking. Motivation for
65 some of the specification decisions is also included.

66 The "Mapping between LPD and IPP Protocols" document gives some advice to implementers of
67 gateways between IPP and LPD (Line Printer Daemon) implementations.

68

Table of Contents

69	1. Introduction	10
70	1.1 Simplified Printing Model	11
71	2. IPP Objects	13
72	2.1 Printer Object	13
73	2.2 Job Object	15
74	2.3 Object Relationships	16
75	2.4 Object Identity	17
76	3. IPP Operations	19
77	3.1 Common Semantics	20
78	3.1.1 Required Parameters	20
79	3.1.2 Operation IDs and Request IDs	21
80	3.1.3 Attributes	21
81	3.1.4 Character Set and Natural Language Operation Attributes	22
82	3.1.4.1 Request Operation Attributes	23
83	3.1.4.2 Response Operation Attributes	26
84	3.1.5 Operation Targets	27
85	3.1.6 Operation Status Codes and Messages	28
86	<u>3.1.7 Unsupported Attributes</u>	<u>29</u>
87	3.1.8 Versions	30
88	3.1.9 Job Creation Operations	31
89	3.2 Printer Operations	33
90	3.2.1 Print-Job Operation	33
91	3.2.1.1 Print-Job Request	33
92	3.2.1.2 Print-Job Response	37
93	3.2.2 Print-URI Operation	39
94	3.2.3 Validate-Job Operation	40
95	3.2.4 Create-Job Operation	40
96	3.2.5 Get-Printer-Attributes Operation	41
97	3.2.5.1 Get-Printer-Attributes Request	42
98	3.2.5.2 Get-Printer-Attributes Response	43
99	3.2.6 Get-Jobs Operation	44
100	3.2.6.1 Get-Jobs Request	44
101	3.2.6.2 Get-Jobs Response	46
102	3.2.7 Pause-Printer Operation	47
103	3.2.7.1 Pause-Printer Request	48
104	3.2.7.2 Pause-Printer Response	49
105	3.2.8 Resume-Printer Operation	49
106	3.2.9 Purge-Jobs Operation	50
107	3.3 Job Operations	50
108	3.3.1 Send-Document Operation	51
109	3.3.1.1 Send-Document Request	52

110	3.3.1.2	Send-Document Response	53
111	3.3.2	Send-URI Operation	54
112	3.3.3	Cancel-Job Operation	54
113	3.3.3.1	Cancel-Job Request	55
114	3.3.3.2	Cancel-Job Response	56
115	3.3.4	Get-Job-Attributes Operation	57
116	3.3.4.1	Get-Job-Attributes Request	57
117	3.3.4.2	Get-Job-Attributes Response	58
118	3.3.5	Hold-Job Operation	59
119	3.3.5.1	Hold-Job Request	60
120	3.3.5.2	Hold-Job Response	60
121	3.3.6	Release-Job Operation	60
122	3.3.7	Restart-Job Operation	61
123	3.3.7.1	Restart-Job Request	62
124	3.3.7.2	Restart-Job Response	63
125	4.	Object Attributes	63
126	4.1	Attribute Syntaxes	64
127	4.1.1	'text'	64
128	4.1.1.1	'textWithoutLanguage'	65
129	4.1.1.2	'textWithLanguage'	65
130	4.1.2	'name'	66
131	4.1.2.1	'nameWithoutLanguage'	66
132	4.1.2.2	'nameWithLanguage'	66
133	4.1.2.3	Matching 'name' attribute values	67
134	4.1.3	'keyword'	67
135	4.1.4	'enum'	68
136	4.1.5	'uri'	69
137	4.1.6	'uriScheme'	69
138	4.1.7	'charset'	69
139	4.1.8	'naturalLanguage'	70
140	4.1.9	'mimeType'	70
141	4.1.9.1	<u>Application/octet-stream -- Auto-Sensing the document format</u>	71
142	4.1.10	'octetString'	72
143	4.1.11	'boolean'	72
144	4.1.12	'integer'	72
145	4.1.13	'rangeOfInteger'	72
146	4.1.14	'dateTime'	72
147	4.1.15	'resolution'	72
148	4.1.16	'setOf X'	73
149	4.2	Job Template Attributes	73
150	4.2.1	job-priority (integer(1:100))	76
151	4.2.2	job-hold-until (type3 keyword name (MAX))	77
152	4.2.3	job-sheets (type3 keyword name(MAX))	77
153	4.2.4	multiple-document-handling (type2 keyword)	78

154	4.2.5	copies (integer(1:MAX))	79
155	4.2.6	finishings (1setOf type2 enum)	79
156	4.2.7	page-ranges (1setOf rangeOfInteger (1:MAX))	81
157	4.2.8	sides (type2 keyword)	82
158	4.2.9	number-up (integer(1:MAX))	82
159	4.2.10	orientation-requested (type2 enum)	83
160	4.2.11	media (type3 keyword name(MAX))	84
161	4.2.12	printer-resolution (resolution)	84
162	4.2.13	print-quality (type2 enum)	84
163	4.3	Job Description Attributes	85
164	4.3.1	job-uri (uri)	87
165	4.3.2	job-id (integer(1:MAX))	87
166	4.3.3	job-printer-uri (uri)	87
167	4.3.4	job-more-info (uri)	87
168	4.3.5	job-name (name(MAX))	88
169	4.3.6	job-originating-user-name (name(MAX))	88
170	4.3.7	job-state (type1 enum)	88
171	4.3.7.1	Partitioning of Job States	91
172	4.3.8	job-state-reasons (1setOf type2 keyword)	91
173	4.3.9	job-state-message (text(MAX))	94
174	4.3.10	number-of-documents (integer(0:MAX))	95
175	4.3.11	output-device-assigned (name(127))	95
176	4.3.12	Event Time Job Description Attributes Issue 17	95
177	4.3.12.1	time-at-creation (integer(MIN:MAX) dateTime)	96
178	4.3.12.2	time-at-processing (integer(MIN:MAX) dateTime)	96
179	4.3.12.3	time-at-completed (integer(MIN:MAX) dateTime)	96
180	4.3.12.4	job-printer-up-time (integer(1:MAX)) Issue 17	96
181	4.3.13	number-of-intervening-jobs (integer(0:MAX))	97
182	4.3.14	job-message-from-operator (text(127))	97
183	4.3.15	Job Size Attributes	97
184	4.3.15.1	job-k-octets (integer(0:MAX))	97
185	4.3.15.2	job-impressions (integer(0:MAX))	98
186	4.3.15.3	job-media-sheets (integer(0:MAX))	98
187	4.3.16	Job Progress Attributes	98
188	4.3.16.1	job-k-octets-processed (integer(0:MAX))	98
189	4.3.16.2	job-impressions-completed (integer(0:MAX))	99
190	4.3.16.3	job-media-sheets-completed (integer(0:MAX))	99
191	4.3.17	attributes-charset (charset)	99
192	4.3.18	attributes-natural-language (naturalLanguage)	99
193	4.4	Printer Description Attributes	100
194	4.4.1	printer-uri-supported (1setOf uri)	102
195	4.4.2	uri-authentication-supported (1setOf type2 keyword) Issue 2	103
196	4.4.3	uri-security-supported (1setOf type2 keyword)	103
197	4.4.4	printer-name (name(127))	105
198	4.4.5	printer-location (text(127))	105

199	4.4.6	printer-info (text(127))	105
200	4.4.7	printer-more-info (uri)	105
201	4.4.8	printer-driver-installer (uri)	105
202	4.4.9	printer-make-and-model (text(127)).....	105
203	4.4.10	printer-more-info-manufacturer (uri)	105
204	4.4.11	printer-state (type1 enum)	106
205	4.4.12	printer-state-reasons (1setOf type2 keyword).....	107
206	4.4.13	printer-state-message (text(MAX))	109
207	<u>4.4.14</u>	<u>ipp-versions-supported (1setOf type2 keyword) Issue 36</u>	109
208	4.4.15	operations-supported (1setOf type2 enum)	110
209	<u>4.4.16</u>	<u>multiple-document-jobs-supported (boolean) Issue 34</u>	111
210	4.4.17	charset-configured (charset)	111
211	4.4.18	charset-supported (1setOf charset)	111
212	4.4.19	natural-language-configured (naturalLanguage)	111
213	4.4.20	generated-natural-language-supported (1setOf naturalLanguage).....	111
214	4.4.21	document-format-default (mimeType).....	112
215	4.4.22	document-format-supported (1setOf mimeType).....	112
216	4.4.23	printer-is-accepting-jobs (boolean).....	112
217	4.4.24	queued-job-count (integer(0:MAX))	112
218	4.4.25	printer-message-from-operator (text(127)).....	113
219	4.4.26	color-supported (boolean)	113
220	4.4.27	reference-uri-schemes-supported (1setOf uriScheme)	113
221	4.4.28	pdl-override-supported (type2 keyword)	113
222	4.4.29	printer-up-time (integer(1:MAX)).....	113
223	4.4.30	printer-current-time (dateTime).....	114
224	4.4.31	multiple-operation-time-out (integer(1:MAX)).....	115
225	4.4.32	compression-supported (1setOf type3 keyword).....	115
226	4.4.33	job-k-octets-supported (rangeOfInteger(0:MAX))	115
227	4.4.34	job-impressions-supported (rangeOfInteger(0:MAX)).....	115
228	4.4.35	job-media-sheets-supported (rangeOfInteger(0:MAX)).....	116
229	4.4.36	pages-per-minute (integer(0:MAX))	116
230	4.4.37	pages-per-minute-color (integer(0:MAX)).....	116
231	5.	Conformance	116
232	5.1	Client Conformance Requirements.....	117
233	5.2	IPP Object Conformance Requirements	118
234	5.2.1	Objects.....	118
235	5.2.2	Operations.....	118
236	5.2.3	IPP Object Attributes.....	119
237	5.2.4	Versions.....	119
238	5.2.5	Extensions	119
239	5.2.6	Attribute Syntaxes	119
240	<u>5.2.7</u>	<u>Security Issue 32</u>	120
241	5.3	Charset and Natural Language Requirements.....	120
242	6.	IANA Considerations (registered and private extensions)	121

243	6.1	Typed 'keyword' and 'enum' Extensions	121
244	6.2	Attribute Extensibility	123
245	6.3	Attribute Syntax Extensibility	123
246	6.4	Operation Extensibility	124
247	6.5	Attribute Groups	124
248	6.6	Status Code Extensibility.....	124
249	6.7	Registration of MIME types/sub-types for document-formats	125
250	6.8	Registration of charsets for use in 'charset' attribute values	125
251	7.	Internationalization Considerations.....	125
252	8.	Security Considerations.....	128
253	8.1	Security Scenarios.....	129
254	8.1.1	Client and Server in the Same Security Domain	129
255	8.1.2	Client and Server in Different Security Domains.....	129
256	8.1.3	Print by Reference	129
257	8.2	URIs in Operation, Job, and Printer attributes.....	130
258	8.3	URIs for each authentication mechanisms.....	130
259	8.4	Restricted Queries.....	132
260	8.5	Operations performed by operators and system administrators.....	132
261	8.6	Queries on jobs submitted using non-IPP protocols.....	132
262	9.	References	133
263	10.	Author's Address	137
264	11.	Formats for IPP Registration Proposals	141
265	11.1	Type2 keyword attribute values registration.....	141
266	11.2	Type3 keyword attribute values registration.....	141
267	11.3	Type2 enum attribute values registration.....	141
268	11.4	Type3 enum attribute values registration.....	142
269	11.5	Attribute registration.....	142
270	11.6	Attribute Syntax registration.....	143
271	11.7	Operation registration	143
272	11.8	Attribute Group registration	143
273	11.9	Status code registration.....	144
274	12.	APPENDIX A: Terminology	144
275	12.1	Conformance Terminology.....	144
276	12.1.1	NEED NOT	144
277	12.2	Model Terminology	144
278	12.2.1	Keyword	144
279	12.2.2	Attributes	144
280	12.2.2.1	Attribute Name	145
281	12.2.2.2	Attribute Group Name	145
282	12.2.2.3	Attribute Value	145

283	12.2.2.4	Attribute Syntax.....	145
284	12.2.3	Supports.....	145
285	12.2.4	print-stream page.....	147
286	12.2.5	impression.....	147
287	13.	APPENDIX B: Status Codes and Suggested Status Code Messages.....	147
288	13.1	Status Codes.....	148
289	13.1.1	Informational.....	148
290	13.1.2	Successful Status Codes.....	148
291	13.1.2.1	successful-ok (0x0000).....	148
292	13.1.2.2	successful-ok-ignored-or-substituted-attributes (0x0001).....	149
293	13.1.2.3	successful-ok-conflicting-attributes (0x0002).....	149
294	13.1.3	Redirection Status Codes.....	149
295	13.1.4	Client Error Status Codes.....	149
296	13.1.4.1	client-error-bad-request (0x0400).....	149
297	13.1.4.2	client-error-forbidden (0x0401).....	150
298	13.1.4.3	client-error-not-authenticated (0x0402).....	150
299	13.1.4.4	client-error-not-authorized (0x0403).....	150
300	13.1.4.5	client-error-not-possible (0x0404).....	150
301	13.1.4.6	client-error-timeout (0x0405).....	150
302	13.1.4.7	client-error-not-found (0x0406).....	150
303	13.1.4.8	client-error-gone (0x0407).....	151
304	13.1.4.9	client-error-request-entity-too-large (0x0408).....	151
305	13.1.4.10	client-error-request-value-too-long (0x0409).....	151
306	13.1.4.11	client-error-document-format-not-supported (0x040A).....	152
307	13.1.4.12	client-error-attributes-or-values-not-supported (0x040B).....	152
308	13.1.4.13	client-error-uri-scheme-not-supported (0x040C).....	152
309	13.1.4.14	client-error-charset-not-supported (0x040D).....	152
310	13.1.4.15	client-error-conflicting-attributes (0x040E).....	153
311	13.1.4.16	<u>client-error-compression-not-supported (0x040F) Issue 6</u>	153
312	13.1.4.17	<u>client-error-compression-error (0x0410) Issue 6</u>	153
313	13.1.4.18	<u>client-error-document-format-error (0x0411) Issue 28</u>	153
314	13.1.4.19	<u>client-error-document-access-error (0x0412) Issue 35</u>	153
315	13.1.5	Server Error Status Codes.....	154
316	13.1.5.1	server-error-internal-error (0x0500).....	154
317	13.1.5.2	server-error-operation-not-supported (0x0501).....	154
318	13.1.5.3	server-error-service-unavailable (0x0502).....	154
319	13.1.5.4	server-error-version-not-supported (0x0503).....	154
320	13.1.5.5	server-error-device-error (0x0504).....	155
321	13.1.5.6	server-error-temporary-error (0x0505).....	155
322	13.1.5.7	server-error-not-accepting-jobs (0x0506).....	155
323	13.1.5.8	server-error-busy (0x0507).....	155
324	13.1.5.9	server-error-job-canceled (0x0508).....	155
325	13.1.5.10	<u>server-error-multiple-document-jobs-not-supported (0x0509) Issue 34</u>	155
326	13.2	Status Codes for IPP Operations.....	156

327 14. APPENDIX C: "media" keyword values 158

328 15. APPENDIX D: Processing IPP Attributes 161

329 15.1 Fidelity 162

330 15.2 Page Description Language (PDL) Override 163

331 15.3 Using Job Template Attributes During Document Processing..... 164

332 16. APPENDIX E: Generic Directory Schema 165

333 17. APPENDIX F: Differences between the IPP/1.0 and IPP/1.1 "Model and Semantics" Specifications

334 168

335 18. Full Copyright Statement 172

336

337 1. Introduction

338 The Internet Printing Protocol (IPP) is an application level protocol that can be used for distributed
339 printing using Internet tools and technologies. IPP version 1.1 (IPP/1.1) focuses only on end user
340 functionality. This document is just one of a suite of documents that fully define IPP. The full set of
341 IPP documents includes:

- 342 Design Goals for an Internet Printing Protocol [[RFC2567](#)~~IPP-REQ~~]
- 343 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [[RFC2568](#)~~IPP-~~
344 ~~RAT~~]
- 345 Internet Printing Protocol/1.1: Model and Semantics (this document)
- 346 Internet Printing Protocol/1.1: Encoding and Transport [IPP-PRO]
- 347 Internet Printing Protocol/1.1: Implementer's Guide [IPP-IIG]
- 348 Mapping between LPD and IPP Protocols [[RFC2569](#)~~IPP-LPD~~]

350 Anyone reading these documents for the first time is strongly encouraged to read the IPP documents in
351 the above order.

352 This document is laid out as follows:

- 353 - The rest of Section 1 is an introduction to the IPP simplified model for distributed printing.
- 354 - Section 2 introduces the object types covered in the model with their basic behaviors, attributes,
355 and interactions.
- 356 - Section 3 defines the operations included in IPP/1.1. IPP operations are synchronous, therefore, for
357 each operation, there is a both request and a response.
- 358 - Section 4 defines the attributes (and their syntaxes) that are used in the model.
- 359 - Sections 5 - 6 summarizes the implementation conformance requirements for objects that support
360 the protocol and IANA considerations, respectively.
- 361 - Sections 7 - 12 cover the Internationalization and Security considerations as well as References,
362 Intellectual Property Notice, Copyright Notice, Author contact information, and Formats for
363 Registration Proposals.
- 364 - Sections 13 - 15 are appendices that cover Terminology, Status Codes and Messages, and "media"
365 keyword values.

366 Note: This document uses terms such as "attributes", "keywords", and "support". These
367 terms have special meaning and are defined in the model terminology section 12.2.
368 Capitalized terms, such as MUST, MUST NOT, REQUIRED, SHOULD, SHOULD
369 NOT, MAY, NEED NOT, and OPTIONAL, have special meaning relating to
370 conformance. These terms are defined in section 12.1 on conformance terminology, most
371 of which is taken from RFC 2119 [RFC2119].

- 372 - Section 16 is an appendix that helps to clarify the effects of interactions between related attributes
373 and their values.
- 374 - Section 17 is an appendix that enumerates the subset of Printer attributes that form a generic
375 directory schema. These attributes are useful when registering a Printer so that a client can find
376 the Printer not just by name, but by filtered searches as well.

377 - Section 18 is an appendix summarizing the additions and changes from the IPP/1.0 "Model and
378 Semantics" specification [~~IPP-MOD1.0~~[RFC2566](#)] to make this IPP/1.1 document.

379 1.1 Simplified Printing Model

380 In order to achieve its goal of realizing a workable printing protocol for the Internet, the Internet Printing
381 Protocol (IPP) is based on a simplified printing model that abstracts the many components of real world
382 printing solutions. The Internet is a distributed computing environment where requesters of print
383 services (clients, applications, printer drivers, etc.) cooperate and interact with print service providers.
384 This model and semantics document describes a simple, abstract model for IPP even though the
385 underlying configurations may be complex "n-tier" client/server systems. An important simplifying step
386 in the IPP model is to expose only the key objects and interfaces required for printing. The model
387 described in this model document does not include features, interfaces, and relationships that are beyond
388 the scope of the first version of IPP (IPP/1.1). IPP/1.1 incorporates many of the relevant ideas and
389 lessons learned from other specification and development efforts [HTPP] [ISO10175] [LDPA]
390 [P1387.4] [PSIS] [RFC1179] [SWP]. IPP is heavily influenced by the printing model introduced in the
391 Document Printing Application (DPA) [ISO10175] standard. Although DPA specifies both end user and
392 administrative features, IPP version 1.1 (IPP/1.1) focuses primarily on end user functionality with a few
393 additional OPTIONAL operator operations.

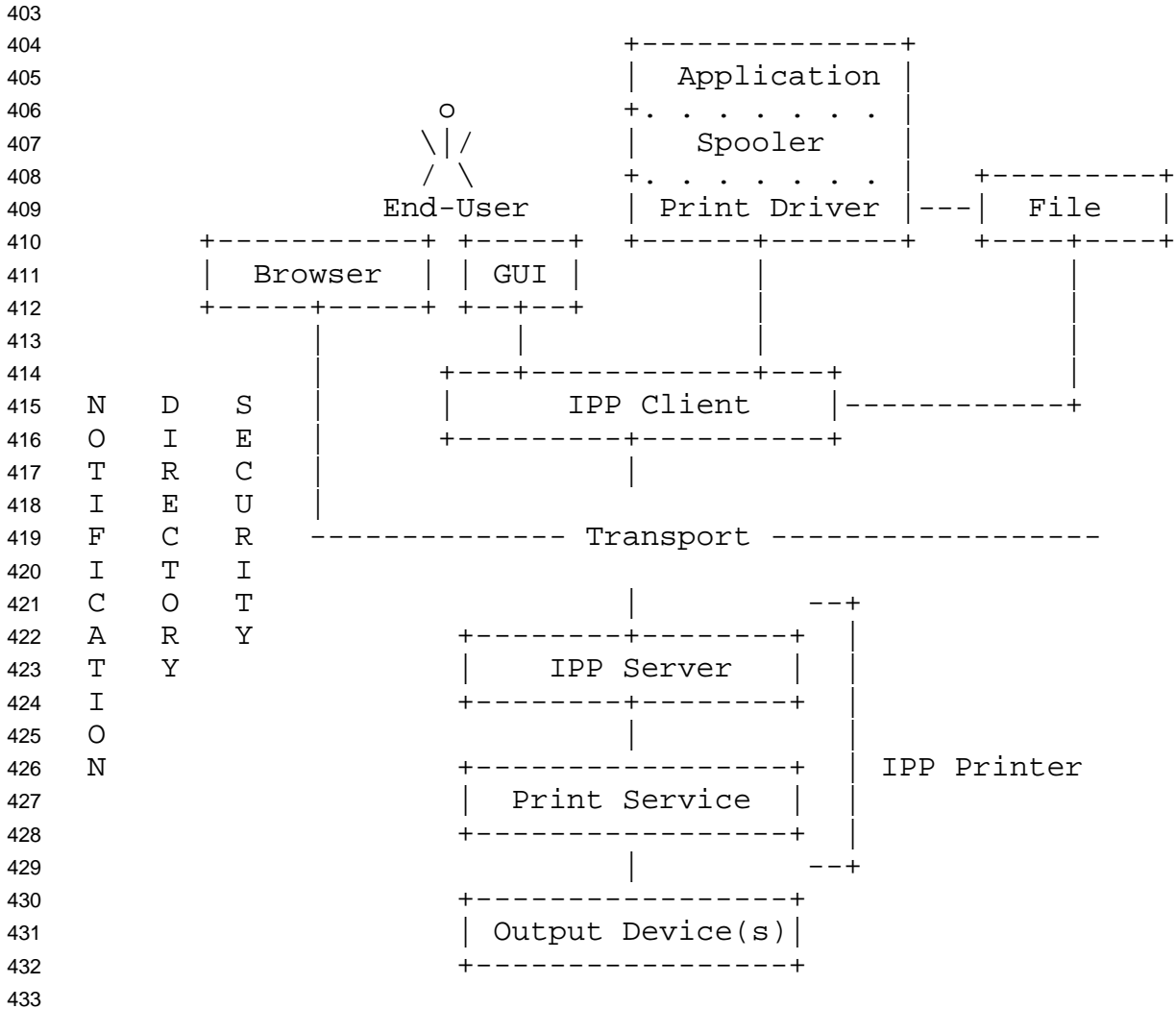
394 The IPP/1.1 model encapsulates the important components of distributed printing into two object types:

- 395 - Printer (Section 2.1)
- 396 - Job (Section 2.2)

397

398 Each object type has an associated set of operations (see section 3) and attributes (see section 3.3.5).

399 It is important, however, to understand that in real system implementations (which lie underneath the
400 abstracted IPP/1.1 model), there are other components of a print service which are not explicitly defined
401 in the IPP/1.1 model. The following figure illustrates where IPP/1.1 fits with respect to these other
402 components.



434 An IPP Printer object encapsulates the functions normally associated with physical output devices along
435 with the spooling, scheduling and multiple device management functions often associated with a print
436 server. Printer objects are optionally registered as entries in a directory where end users find and select
437 them based on some sort of filtered and context based searching mechanism (see section 16). The
438 directory is used to store relatively static information about the Printer, allowing end users to search for
439 and find Printers that match their search criteria, for example: name, context, printer capabilities, etc.
440 The more dynamic information, such as state, currently loaded and ready media, number of jobs at the
441 Printer, errors, warnings, and so forth, is directly associated with the Printer object itself rather than with
442 the entry in the directory which only represents the Printer object.

443 IPP clients implement the IPP protocol on the client side and give end users (or programs running on
444 behalf of end users) the ability to query Printer objects and submit and manage print jobs. An IPP server
445 is just that part of the Printer object that implements the server-side protocol. The rest of the Printer
446 object implements (or gateways into) the application semantics of the print service itself. The Printer
447 objects may be embedded in an output device or may be implemented on a host on the network that
448 communicates with an output device.

449 When a job is submitted to the Printer object and the Printer object validates the attributes in the
450 submission request, the Printer object creates a new Job object. The end user then interacts with this
451 new Job object to query its status and monitor the progress of the job. An end user can also cancel their
452 print jobs by using the Job object's Cancel-Job operation. An end-user can also hold, release, and restart
453 their print jobs using the Job object's OPTIONAL Hold-Job, Release-Job, and Restart-Job operations, if
454 implemented.

455 A privileged operator or administrator of a Printer object can cancel, hold, release, and restart any user's
456 job using the REQUIRED Cancel-Job and the OPTIONAL Hold-Job, Release-Job, and Restart-Job
457 operations. In addition, a privileged operator or administrator of a Printer object can pause, resume, or
458 purge (jobs from) a Printer object using the OPTIONAL Pause-Printer, Resume-Printer, and Purge-Jobs
459 operations, if implemented.

460 The notification service is out of scope for this IPP/1.1 specification, but using such a notification
461 service, the end user is able to register for and receive Printer specific and Job specific events. An end
462 user can query the status of Printer objects and can follow the progress of Job objects by polling using
463 the Get-Printer-Attributes, Get-Jobs, and Get-Job-Attributes operations.

464 2. IPP Objects

465 The IPP/1.1 model introduces objects of type Printer and Job. Each type of object models relevant
466 aspects of a real-world entity such as a real printer or real print job. Each object type is defined as a set
467 of possible attributes that may be supported by instances of that object type. For each object (instance),
468 the actual set of supported attributes and values describe a specific implementation. The object's
469 attributes and values describe its state, capabilities, realizable features, job processing functions, and
470 default behaviors and characteristics. For example, the Printer object type is defined as a set of attributes
471 that each Printer object potentially supports. In the same manner, the Job object type is defined as a set
472 of attributes that are potentially supported by each Job object.

473 Each attribute included in the set of attributes defining an object type is labeled as:

- 474 - "REQUIRED": each object MUST support the attribute.
- 475 - "OPTIONAL": each object MAY support the attribute.

476

477 There is no such similar labeling of attribute values. However, if an implementation supports an
478 attribute, it MUST support at least one of the possible values for that attribute.

479 2.1 Printer Object

480 The major component of the IPP/1.1 model is the Printer object. A Printer object implements the server-
481 side of the IPP/1.1 protocol. Using the protocol, end users may query the attributes of the Printer object
482 and submit print jobs to the Printer object. The actual implementation components behind the Printer
483 abstraction may take on different forms and different configurations. However, the model abstraction

484 allows the details of the configuration of real components to remain opaque to the end user. Section 3
485 describes each of the Printer operations in detail.

486 The capabilities and state of a Printer object are described by its attributes. Printer attributes are divided
487 into two groups:

- 488 - "job-template" attributes: These attributes describe supported job processing capabilities and
489 defaults for the Printer object. (See section 4.2)
 - 490 - "printer-description" attributes: These attributes describe the Printer object's identification, state,
491 location, references to other sources of information about the Printer object, etc. (see section 4.4)
- 492

493 Since a Printer object is an abstraction of a generic document output device and print service provider, a
494 Printer object could be used to represent any real or virtual device with semantics consistent with the
495 Printer object, such as a fax device, an imager, or even a CD writer.

496 Some examples of configurations supporting a Printer object include:

- 497 1) An output device with no spooling capabilities
 - 498 2) An output device with a built-in spooler
 - 499 3) A print server supporting IPP with one or more associated output devices
 - 500 3a) The associated output devices may or may not be capable of spooling jobs
 - 501 3b) The associated output devices may or may not support IPP
- 502

503 The following figures show some examples of how Printer objects can be realized on top of various
504 distributed printing configurations. The embedded case below represents configurations 1 and 2. The
505 hosted and fan-out figures below represent configurations 3a and 3b.

506 In this document the term "client" refers to a software entity that sends IPP operation request to an IPP
507 Printer object and accepts IPP operation responses. A client MAY be:

- 508 1. contained within software controlled by an end user, e.g. activated by the "Print" menu item in an
509 application and/or
- 510 2. a component of a print server that communicates (using IPP operations) with either an output
511 device or another "downstream" print server.

512 The term "IPP Printer" is a network entity that accepts IPP operation requests and returns IPP operation
513 responses. As such, an IPP object MAY be:

- 514 1. (embedded) software that controls a device
- 515 2. part of a print server that accepts IPP operation requests and, in turn, sends operation requests
516 using (the IPP or other) protocol to one or more networked device(s).

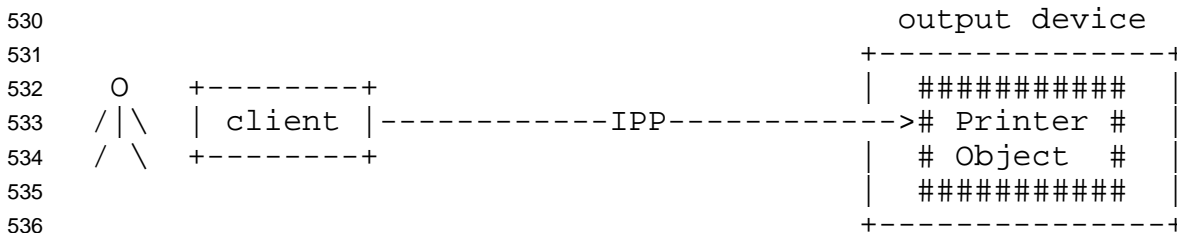
517

518 Legend:

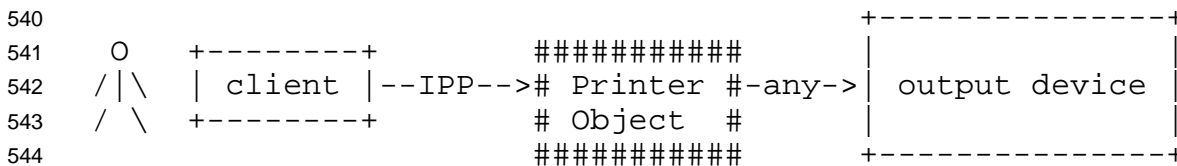
519
520 ##### indicates a Printer object which is
521 either embedded in an output device or is
522 hosted in a server. The Printer object
523 might or might not be capable of queuing/spooling.
524

525 any indicates any network protocol or direct
526 connect, including IPP
527

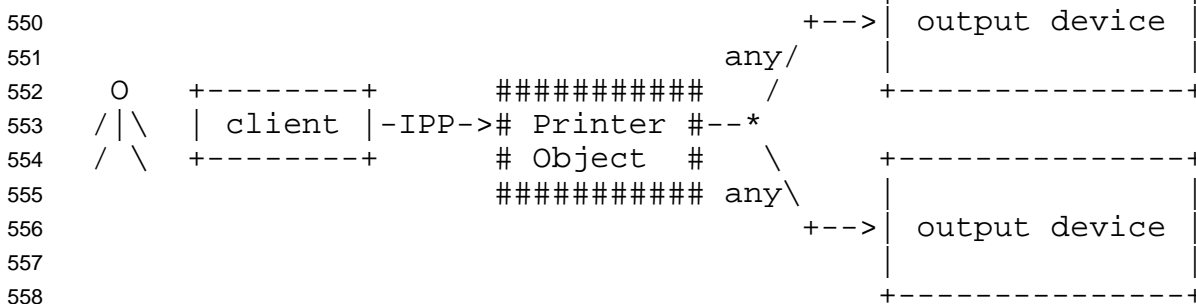
528
529 embedded printer:



539 hosted printer:



548
549 fan out:



561 2.2 Job Object

562 A Job object is used to model a print job. A Job object contains documents. The information required
563 to create a Job object is sent in a create request from the end user via an IPP Client to the Printer object.

564 The Printer object validates the create request, and if the Printer object accepts the request, the Printer
565 object creates the new Job object. Section 3 describes each of the Job operations in detail.

566 The characteristics and state of a Job object are described by its attributes. Job attributes are grouped
567 into two groups as follows:

- 568 - "job-template" attributes: These attributes can be supplied by the client or end user and include job
569 processing instructions which are intended to override any Printer object defaults and/or
570 instructions embedded within the document data. (See section 4.2)
- 571 - "job-description" attributes: These attributes describe the Job object's identification, state, size, etc.
572 The client supplies some of these attributes, and the Printer object generates others. (See section
573 4.3)

574
575 An implementation **MUST** support at least one document per Job object. An implementation **MAY**
576 support multiple documents per Job object. A document is either:

- 577 - a stream of document data in a format supported by the Printer object (typically a Page Description
578 Language - PDL), or
- 579 - a reference to such a stream of document data

580

581 In IPP/1.1, a document is not modeled as an IPP object, therefore it has no object identifier or associated
582 attributes. All job processing instructions are modeled as Job object attributes. These attributes are
583 called Job Template attributes and they apply equally to all documents within a Job object.

584 2.3 Object Relationships

585 IPP objects have relationships that are maintained persistently along with the persistent storage of the
586 object attributes.

587 A Printer object can represent either one or more physical output devices or a logical device which
588 "processes" jobs but never actually uses a physical output device to put marks on paper. Examples of
589 logical devices include a Web page publisher or a gateway into an online document archive or
590 repository. A Printer object contains zero or more Job objects.

591 A Job object is contained by exactly one Printer object, however the identical document data associated
592 with a Job object could be sent to either the same or a different Printer object. In this case, a second Job
593 object would be created which would be almost identical to the first Job object, however it would have
594 new (different) Job object identifiers (see section 2.4).

595 A Job object is either empty (before any documents have been added) or contains one or more
596 documents. If the contained document is a stream of document data, that stream can be contained in
597 only one document. However, there can be identical copies of the stream in other documents in the same
598 or different Job objects. If the contained document is just a reference to a stream of document data,
599 other documents (in the same or different Job object(s)) may contain the same reference.

600 2.4 Object Identity

601 All Printer and Job objects are identified by a Uniform Resource Identifier (URI) [RFC2396] so that they
602 can be persistently and unambiguously referenced. The notion of a URI is a useful concept, however,
603 until the notion of URI is more stable (i.e., defined more completely and deployed more widely), it is
604 expected that the URIs used for IPP objects will actually be URLs [RFC2396]. Since every URL is a
605 specialized form of a URI, even though the more generic term URI is used throughout the rest of this
606 document, its usage is intended to cover the more specific notion of URL as well.

607 An administrator configures Printer objects to either support or not support authentication and/or
608 message privacy using TLS [TLS] (the mechanism for security configuration is outside the scope of this
609 IPP/1.1 document). In some situations, both types of connections (both authenticated and
610 unauthenticated) can be established using a single communication channel that has some sort of
611 negotiation mechanism. In other situations, multiple communication channels are used, one for each
612 type of security configuration. Section 8 provides a full description of all security considerations and
613 configurations.

614 If a Printer object supports more than one communication channel, some or all of those channels might
615 support and/or require different security mechanisms. In such cases, an administrator could expose the
616 simultaneous support for these multiple communication channels as multiple URIs for a single Printer
617 object where each URI represents one of the communication channels to the Printer object. To support
618 this flexibility, the IPP Printer object type defines a multi-valued identification attribute called the
619 "printer-uri-supported" attribute. It MUST contain at least one URI. It MAY contain more than one
620 URI. That is, every Printer object will have at least one URI that identifies at least one communication
621 channel to the Printer object, but it may have more than one URI where each URI identifies a different
622 communication channel to the Printer object. The "printer-uri-supported" attribute has ~~a two~~ companion
623 attributes, the "uri-security-supported" attribute and the "uri-authentication-supported"; ~~that has Both~~
624 have the same cardinality as "printer-uri-supported". The purpose of the "uri-security-supported"
625 attribute is to indicate the security mechanisms (if any) used for each URI listed in "printer-uri-
626 supported". The purpose of the "uri-authentication-supported" attribute is to indicate the authentication
627 mechanisms (if any) used for each URI listed in "printer-uri-supported". ~~These two three~~ attributes are
628 fully described in sections 4.4.1, ~~and~~ 4.4.2, ~~and~~ 4.4.3.4.2. **Issue 2**

629 When a job is submitted to the Printer object via a create request, the client supplies only a single Printer
630 object URI. The client supplied Printer object URI MUST be one of the values in the "printer-uri-
631 supported" Printer attribute.

632 Note: IPP/1.1 does not specify how the client obtains the client supplied URI, but it is
633 RECOMMENDED that a Printer object be registered as an entry in a directory service. End-users and
634 programs can then interrogate the directory searching for Printers. Section 16 defines a generic schema
635 for Printer object entries in the directory service and describes how the entry acts as a bridge to the actual
636 IPP Printer object. The entry in the directory that represents the IPP Printer object includes the possibly
637 many URIs for that Printer object as values in one its attributes.

638 When a client submits a create request to the Printer object, the Printer object validates the request and
639 creates a new Job object. The Printer object assigns the new Job object a URI which is stored in the

640 "job-uri" Job attribute. This URI is then used by clients as the target for subsequent Job operations. The
641 Printer object generates a Job URI based on its configured security policy and the URI used by the client
642 in the create request.

643 For example, consider a Printer object that supports both a communication channel secured by the use of
644 SSL3 (using HTTP over SSL3 with an "https" schemed URI) and another open communication channel
645 that is not secured with SSL3 (using a simple "http" schemed URI). If a client were to submit a job
646 using the secure URI, the Printer object would assign the new Job object a secure URI as well. If a client
647 were to submit a job using the open-channel URI, the Printer would assign the new Job object an open-
648 channel URI.

649 In addition, the Printer object also populates the Job object's "job-printer-uri" attribute. This is a
650 reference back to the Printer object that created the Job object. If a client only has access to a Job
651 object's "job-uri" identifier, the client can query the Job's "job-printer-uri" attribute in order to determine
652 which Printer object created the Job object. If the Printer object supports more than one URI, the Printer
653 object picks the one URI supplied by the client when creating the job to build the value for and to
654 populate the Job's "job-printer-uri" attribute.

655 Allowing Job objects to have URIs allows for flexibility and scalability. For example, in some
656 implementations, the Printer object might create Jobs that are processed in the same local environment
657 as the Printer object itself. In this case, the Job URI might just be a composition of the Printer's URI and
658 some unique component for the Job object, such as the unique 32-bit positive integer mentioned later in
659 this paragraph. In other implementations, the Printer object might be a central clearing-house for
660 validating all Job object creation requests, but the Job object itself might be created in some environment
661 that is remote from the Printer object. In this case, the Job object's URI may have no physical-location
662 relationship at all to the Printer object's URI. Again, the fact that Job objects have URIs allows for
663 flexibility and scalability, however, many existing printing systems have local models or interface
664 constraints that force print jobs to be identified using only a 32-bit positive integer rather than an
665 independent URI. This numeric Job ID is only unique within the context of the Printer object to which
666 the create request was originally submitted. Therefore, in order to allow both types of client access to
667 IPP Job objects (either by Job URI or by numeric Job ID), when the Printer object successfully processes
668 a create request and creates a new Job object, the Printer object **MUST** generate both a Job URI and a
669 Job ID. The Job ID (stored in the "job-id" attribute) only has meaning in the context of the Printer object
670 to which the create request was originally submitted. This requirement to support both Job URIs and Job
671 IDs allows all types of clients to access Printer objects and Job objects no matter the local constraints
672 imposed on the client implementation.

673 In addition to identifiers, Printer objects and Job objects have names ("printer-name" and "job-name").
674 An object name **NEED NOT** be unique across all instances of all objects. A Printer object's name is
675 chosen and set by an administrator through some mechanism outside the scope of this IPP/1.1 document.
676 A Job object's name is optionally chosen and supplied by the IPP client submitting the job. If the client
677 does not supply a Job object name, the Printer object generates a name for the new Job object. In all
678 cases, the name only has local meaning.

679 To summarize:

- 680 - Each Printer object is identified with one or more URIs. The Printer's "printer-uri-supported"
681 attribute contains the URI(s).
- 682 - The Printer object's "uri-security-supported" attribute identifies the communication channel security
683 protocols that may or may not have been configured for the various Printer object URIs (e.g., 'tls'
684 or 'none').
- 685 - The Printer object's "uri-authentication-supported" attribute identifies the authentication
686 mechanisms that may or may not have been configured for the various Printer object URIs (e.g.,
687 'digest' or 'none').
- 688 Each Job object is identified with a Job URI. The Job's "job-uri" attribute contains the URI.
- 689 - Each Job object is also identified with Job ID which is a 32-bit, positive integer. The Job's "job-id"
690 attribute contains the Job ID. The Job ID is only unique within the context of the Printer object
691 which created the Job object.
- 692 - Each Job object has a "job-printer-uri" attribute which contains the URI of the Printer object that
693 was used to create the Job object. This attribute is used to determine the Printer object that
694 created a Job object when given only the URI for the Job object. This linkage is necessary to
695 determine the languages, charsets, and operations which are supported on that Job (the basis for
696 such support comes from the creating Printer object).
- 697 - Each Printer object has a name (which is not necessarily unique). The administrator chooses and
698 sets this name through some mechanism outside the scope of this IPP/1.1 document. The Printer
699 object's "printer-name" attribute contains the name.
- 700 - Each Job object has a name (which is not necessarily unique). The client optionally supplies this
701 name in the create request. If the client does not supply this name, the Printer object generates a
702 name for the Job object. The Job object's "job-name" attribute contains the name.

703 3. IPP Operations

704 IPP objects support operations. An operation consists of a request and a response. When a client
705 communicates with an IPP object, the client issues an operation request to the URI for that object.
706 Operation requests and responses have parameters that identify the operation. Operations also have
707 attributes that affect the run-time characteristics of the operation (the intended target, localization
708 information, etc.). These operation-specific attributes are called operation attributes (as compared to
709 object attributes such as Printer object attributes or Job object attributes). Each request carries along
710 with it any operation attributes, object attributes, and/or document data required to perform the
711 operation. Each request requires a response from the object. Each response indicates success or failure
712 of the operation with a status code as a response parameter. The response contains any operation
713 attributes, object attributes, and/or status messages generated during the execution of the operation
714 request.

715 This section describes the semantics of the IPP operations, both requests and responses, in terms of the
716 parameters, attributes, and other data associated with each operation.

717 The IPP/1.1 Printer operations are:

- 718 Print-Job (section 3.2.1)
719 Print-URI (section 3.2.2)

720 Validate-Job (section 3.2.3)
721 Create-Job (section 3.2.4)
722 Get-Printer-Attributes (section 3.2.5)
723 Get-Jobs (section 3.2.6)
724 Pause-Printer (section 3.3.5)
725 Resume-Printer (section 3.3.6)
726 Purge-Jobs (section 3.3.7)
727

728 The Job operations are:

729 Send-Document (section 3.3.1)
730 Send-URI (section 3.3.2)
731 Cancel-Job (section 3.3.3)
732 Get-Job-Attributes (section 3.3.4)
733 Hold-Job (section 3.3.5)
734 Release-Job (section 3.3.6)
735 Restart-Job (section 3.3.7)
736

737 The Send-Document and Send-URI Job operations are used to add a new document to an existing multi-
738 document Job object created using the Create-Job operation.

739 3.1 Common Semantics

740 All IPP operations require some common parameters and operation attributes. These common elements
741 and their semantic characteristics are defined and described in more detail in the following sections.

742 3.1.1 Required Parameters

743 Every operation request contains the following REQUIRED parameters:

- 744 - a "version-number",
 - 745 - an "operation-id",
 - 746 - a "request-id", and
 - 747 - the attributes that are REQUIRED for that type of request.
- 748

749 Every operation response contains the following REQUIRED parameters:

- 750 - a "version-number",
 - 751 - a "status-code",
 - 752 - the "request-id" that was supplied in the corresponding request, and
 - 753 - the attributes that are REQUIRED for that type of response.
- 754

755 The "Encoding and Transport document [IPP-PRO] defines special rules for the encoding of these
756 parameters. All other operation elements are represented using the more generic encoding rules for
757 attributes and groups of attributes.

758 3.1.2 Operation IDs and Request IDs

759 Each IPP operation request includes an identifying "operation-id" value. Valid values are defined in the
760 "operations-supported" Printer attribute section (see section 4.4.15). The client specifies which
761 operation is being requested by supplying the correct "operation-id" value.

762 In addition, every invocation of an operation is identified by a "request-id" value. For each request, the
763 client chooses the "request-id" which **MUST** be an integer (possibly unique depending on client
764 requirements) in the range from 1 to $2^{31} - 1$ (inclusive). This "request-id" allows clients to manage
765 multiple outstanding requests. The receiving IPP object copies all 32-bits of the client-supplied "request-
766 id" attribute into the response so that the client can match the response with the correct outstanding
767 request, even if the "request-id" is out of range. If the request is terminated before the complete
768 "request-id" is received, the IPP object rejects the request and returns a response with a "request-id" of 0.

769 Note: In some cases, the transport protocol underneath IPP might be a connection oriented protocol that
770 would make it impossible for a client to receive responses in any order other than the order in which the
771 corresponding requests were sent. In such cases, the "request-id" attribute would not be essential for
772 correct protocol operation. However, in other mappings, the operation responses can come back in any
773 order. In these cases, the "request-id" would be essential.

774 3.1.3 Attributes

775 Operation requests and responses are both composed of groups of attributes and/or document data. The
776 attributes groups are:

- 777 - Operation Attributes: These attributes are passed in the operation and affect the IPP object's
778 behavior while processing the operation request and may affect other attributes or groups of
779 attributes. Some operation attributes describe the document data associated with the print job
780 and are associated with new Job objects, however most operation attributes do not persist beyond
781 the life of the operation. The description of each operation attribute includes conformance
782 statements indicating which operation attributes are **REQUIRED** and which are **OPTIONAL** for
783 an IPP object to support and which attributes a client **MUST** supply in a request and an IPP
784 object **MUST** supply in a response.
- 785 - Job Template Attributes: These attributes affect the processing of a job. A client **OPTIONALLY**
786 supplies Job Template Attributes in a create request, and the receiving object **MUST** be prepared
787 to receive all supported attributes. The Job object can later be queried to find out what Job
788 Template attributes were originally requested in the create request, and such attributes are
789 returned in the response as Job Object Attributes. The Printer object can be queried about its Job
790 Template attributes to find out what type of job processing capabilities are supported and/or what
791 the default job processing behaviors are, though such attributes are returned in the response as
792 Printer Object Attributes. The "ipp-attribute-fidelity" operation attribute affects processing of all
793 client-supplied Job Template attributes (see sections 3.2.1.2 and 15 for a full description of "ipp-
794 attribute-fidelity" and its relationship to other attributes).
- 795 - Job Object Attributes: These attributes are returned in response to a query operation directed at a
796 Job object.

- 797 - Printer Object Attributes: These attributes are returned in response to a query operation directed at a
798 Printer object.
- 799 - Unsupported Attributes: In a create request, the client supplies a set of Operation and Job Template
800 attributes. If any of these attributes or their values is unsupported by the Printer object, the
801 Printer object returns the set of unsupported attributes in the response. Sections 3.1.7, 3.2.1.2,
802 and 15 give a full description of how Job Template attributes supplied by the client in a create
803 request are processed by the Printer object and how unsupported attributes are returned to the
804 client. Because of extensibility, any IPP object might receive a request that contains new or
805 unknown attributes or values for which it has no support. In such cases, the IPP object processes
806 what it can and returns the unsupported attributes in the response. This group can occur in the
807 response for other operations as well. The Unsupported Attribute group is defined for all operation
808 responses for returning unsupported attributes that the client supplied in the request. Issue
809

810 Later in this section, each operation is formally defined by identifying the allowed and expected groups
811 of attributes for each request and response. The model identifies a specific order for each group in each
812 request or response, but the attributes within each group may be in any order, unless specified otherwise.

813 Each attribute specification includes the attribute's name followed by the name of its attribute syntax(es)
814 in parentheses. In addition, each 'integer' attribute is followed by the allowed range in parentheses,
815 (m:n), for values of that attribute. Each 'text' or 'name' attribute is followed by the maximum size in
816 octets in parentheses, (size), for values of that attribute. For more details on attribute syntax notation, see
817 the descriptions of these attributes syntaxes in section 4.1.

818 Note: Document data included in the operation is not strictly an attribute, but it is treated as a special
819 attribute group for ordering purposes. The only operations that support supplying the document data
820 within an operation request are Print-Job and Send-Document. There are no operation responses that
821 include document data.

822 Note: Some operations are REQUIRED for IPP objects to support; the others are OPTIONAL (see
823 section 5.2.2). Therefore, before using an OPTIONAL operation, a client SHOULD first use the
824 REQUIRED Get-Printer-Attributes operation to query the Printer's "operations-supported" attribute in
825 order to determine which OPTIONAL Printer and Job operations are actually supported. The client
826 SHOULD NOT use an OPTIONAL operation that is not supported. When an IPP object receives a
827 request to perform an operation it does not support, it returns the 'server-error-operation-not-supported'
828 status code (see section 13.1.5.2). An IPP object is non-conformant if it does not support a REQUIRED
829 operation.

830 3.1.4 Character Set and Natural Language Operation Attributes

831 Some Job and Printer attributes have values that are text strings and names intended for human
832 understanding rather than machine understanding (see the 'text' and 'name' attribute syntax descriptions
833 in section 4.1). The following sections describe two special Operation Attributes called "attributes-
834 charset" and "attributes-natural-language". These attributes are always part of the Operation Attributes
835 group. For most attribute groups, the order of the attributes within the group is not important. However,
836 for these two attributes within the Operation Attributes group, the order is critical. The "attributes-

837 charset" attribute MUST be the first attribute in the group and the "attributes-natural-language" attribute
838 MUST be the second attribute in the group. In other words, these attributes MUST be supplied in every
839 IPP request and response, they MUST come first in the group, and MUST come in the specified order.
840 For job creation operations, the IPP Printer implementation saves these two attributes with the new Job
841 object as Job Description attributes. For the sake of brevity in this document, these operation attribute
842 descriptions are not repeated with every operation request and response, but have a reference back to this
843 section instead.

844 3.1.4.1 Request Operation Attributes

845 The client MUST supply and the Printer object MUST support the following REQUIRED operation
846 attributes in every IPP/1.1 operation request:

847 "attributes-charset" (charset):

848 This operation attribute identifies the charset (coded character set and encoding method) used by
849 any 'text' and 'name' attributes that the client is supplying in this request. It also identifies the
850 charset that the Printer object MUST use (if supported) for all 'text' and 'name' attributes and
851 status messages that the Printer object returns in the response to this request. See Sections 4.1.1
852 and 4.1.2 for the specification of the 'text' and 'name' attribute syntaxes.

853
854 All clients and IPP objects MUST support the 'utf-8' charset [RFC2279] and MAY support
855 additional charsets provided that they are registered with IANA [IANA-CS]. If the Printer object
856 does not support the client supplied charset value, the Printer object MUST reject the request, set
857 the "attributes-charset" to 'utf-8' in the response, and return the 'client-error-charset-not-
858 supported' status code and any 'text' or 'name' attributes using the 'utf-8' charset. [The Printer
859 NEED NOT return any attributes in the Unsupported Attributes Group \(See sections 3.1.7 and
860 3.2.1.2\).](#) The Printer object MUST indicate the charset(s) supported as the values of the "charset-
861 supported" Printer attribute (see Section 4.4.18), so that the client can query to determine which
862 charset(s) are supported.

863
864 Note to client implementers: Since IPP objects are only required to support the 'utf-8' charset, in
865 order to maximize interoperability with multiple IPP object implementations, a client may want
866 to supply 'utf-8' in the "attributes-charset" operation attribute, even though the client is only
867 passing and able to present a simpler charset, such as US-ASCII or ISO-8859-1. Then the client
868 will have to filter out (or charset convert) those characters that are returned in the response that it
869 cannot present to its user. On the other hand, if both the client and the IPP objects also support a
870 charset in common besides utf-8, the client may want to use that charset in order to avoid charset
871 conversion or data loss.

872
873 See the 'charset' attribute syntax description in Section 4.1.7 for the syntax and semantic
874 interpretation of the values of this attribute and for example values.

875
876 "attributes-natural-language" (naturalLanguage):

877 This operation attribute identifies the natural language used by any 'text' and 'name' attributes that
878 the client is supplying in this request. This attribute also identifies the natural language that the

879 Printer object SHOULD use for all 'text' and 'name' attributes and status messages that the Printer
880 object returns in the response to this request.

881
882 There are no REQUIRED natural languages required for the Printer object to support. However,
883 the Printer object's "generated-natural-language-supported" attribute identifies the natural
884 languages supported by the Printer object and any contained Job objects for all text strings
885 generated by the IPP object. A client MAY query this attribute to determine which natural
886 language(s) are supported for generated messages.

887
888 For any of the attributes for which the Printer object generates text, i.e., for the "job-state-
889 message", "printer-state-message", and status messages (see Section 3.1.6), the Printer object
890 MUST be able to generate these text strings in any of its supported natural languages. If the
891 client requests a natural language that is not supported, the Printer object MUST return these
892 generated messages in the Printer's configured natural language as specified by the Printer's
893 "natural-language-configured" attribute" (see Section 4.4.19).

894
895 For other 'text' and 'name' attributes supplied by the client, authentication system, operator,
896 system administrator, or manufacturer (i.e., for "job-originating-user-name", "printer-name"
897 (name), "printer-location" (text), "printer-info" (text), and "printer-make-and-model" (text)), the
898 Printer object is only required to support the configured natural language of the Printer identified
899 by the Printer object's "natural-language-configured" attribute, though support of additional
900 natural languages for these attributes is permitted.

901
902 For any 'text' or 'name' attribute in the request that is in a different natural language than the value
903 supplied in the "attributes-natural-language" operation attribute, the client MUST use the Natural
904 Language Override mechanism (see sections 4.1.1.2 and 4.1.2.2) for each such attribute value
905 supplied. The client MAY use the Natural Language Override mechanism redundantly, i.e., use
906 it even when the value is in the same natural language as the value supplied in the "attributes-
907 natural-language" operation attribute of the request.

908
909 The IPP object MUST accept any natural language and any Natural Language Override, whether
910 the IPP object supports that natural language or not (and independent of the value of the "ipp-
911 attribute-fidelity" Operation attribute). That is the IPP object accepts all client supplied values no
912 matter what the values are in the Printer object's "generated-natural-language-supported"
913 attribute. That attribute, "generated-natural-language-supported", only applies to generated
914 messages, not client supplied messages. The IPP object MUST remember that natural language
915 for all client-supplied attributes, and when returning those attributes in response to a query, the
916 IPP object MUST indicate that natural language.

917
918 Each value whose attribute syntax type is 'text' or 'name' (see sections 4.1.1 and 4.1.2) has an
919 Associated Natural-Language. This document does not specify how this association is stored in a
920 Printer or Job object. When such a value is encoded in a request or response, the natural
921 language is either implicit or explicit:

922

- 923 • — In the implicit case, the value contains only the text/name value, and the language is
924 specified by the "attributes-natural-language" operation attribute in the request or
925 response (see sections 4.1.1.1 textWithoutLanguage and 4.1.2.1 nameWithoutLanguage).
926
- 927 • — In the explicit case (also known as the Natural-Language Override case), the value
928 contains both the language and the text/name value (see sections 4.1.1.2
929 textWithLanguage and 4.1.2.2 nameWithLanguage).
930

931 For example, the "job-name" attribute MAY be supplied by the client in a create request. The
932 text value for this attribute will be in the natural language identified by the "attribute-natural-
933 language" attribute, or if different, as identified by the Natural Language Override mechanism. If
934 supplied, the IPP object will use the value of the "job-name" attribute to populate the Job object's
935 "job-name" attribute. Whenever any client queries the Job object's "job-name" attribute, the IPP
936 object returns the attribute as stored and uses the Natural Language Override mechanism to
937 specify the natural language, if it is different from that reported in the "attributes-natural-
938 language" operation attribute of the response. The IPP object MAY use the Natural Language
939 Override mechanism redundantly, i.e., use it even when the value is in the same natural language
940 as the value supplied in the "attributes-natural-language" operation attribute of the response.
941

942 An IPP object MUST NOT reject a request based on a supplied natural language in an
943 "attributes-natural-language" Operation attribute or in any attribute that uses the Natural
944 Language Override.
945

946 See the 'naturalLanguage' attribute syntax description in section 4.1.8 for the syntax and semantic
947 interpretation of the values of this attribute and for example values.
948

949 Clients SHOULD NOT supply 'text' or 'name' attributes that use an illegal combination of natural
950 language and charset. For example, suppose a Printer object supports charsets 'utf-8', 'iso-8859-1', and
951 'iso-8859-7'. Suppose also, that it supports natural languages 'en' (English), 'fr' (French), and 'el' (Greek).
952 Although the Printer object supports the charset 'iso-8859-1' and natural language 'el', it probably does
953 not support the combination of Greek text strings using the 'iso-8859-1' charset. The Printer object
954 handles this apparent incompatibility differently depending on the context in which it occurs:

- 955 - In a create request: If the client supplies a text or name attribute (for example, the "job-name"
956 operation attribute) that uses an apparently incompatible combination, it is a client choice that
957 does not affect the Printer object or its correct operation. Therefore, the Printer object simply
958 accepts the client supplied value, stores it with the Job object, and responds back with the same
959 combination whenever the client (or any client) queries for that attribute.
- 960 -In a query-type operation, like Get-Printer-Attributes: If the client requests an apparently
961 incompatible combination, the Printer object responds (as described in section 3.1.4.2) using the
962 Printer's configured natural language rather than the natural language requested by the client.
963

964 In either case, the Printer object does not reject the request because of the apparent incompatibility. The
965 potential incompatible combination of charset and natural language can occur either at the global
966 operation level or at the Natural Language Override attribute-by-attribute level. In addition, since the

967 response always includes explicit charset and natural language information, there is never any question
968 or ambiguity in how the client interprets the response.

969 3.1.4.2 Response Operation Attributes

970 The Printer object **MUST** supply and the client **MUST** support the following **REQUIRED** operation
971 attributes in every IPP/1.1 operation response:

972 "attributes-charset" (charset):

973 This operation attribute identifies the charset used by any 'text' and 'name' attributes that the
974 Printer object is returning in this response. The value in this response **MUST** be the same value
975 as the "attributes-charset" operation attribute supplied by the client in the request. If this is not
976 possible (i.e., the charset requested is not supported), the request would have been rejected. See
977 "attributes-charset" described in Section 3.1.4.1 above.

978

979 If the Printer object supports more than just the 'utf-8' charset, the Printer object **MUST** be able to
980 code convert between each of the charsets supported on a highest fidelity possible basis in order
981 to return the 'text' and 'name' attributes in the charset requested by the client. However, some
982 information loss **MAY** occur during the charset conversion depending on the charsets involved.
983 For example, the Printer object may convert from a UTF-8 'a' to a US-ASCII 'a' (with no loss of
984 information), from an ISO Latin 1 CAPITAL LETTER A WITH ACUTE ACCENT to US-
985 ASCII 'A' (losing the accent), or from a UTF-8 Japanese Kanji character to some ISO Latin 1
986 error character indication such as '?', decimal code equivalent, or to the absence of a character,
987 depending on implementation.

988

989 Note: Whether an implementation that supports more than one charset stores the data in the
990 charset supplied by the client or code converts to one of the other supported charsets, depends on
991 implementation. The strategy should try to minimize loss of information during code conversion.
992 On each response, such an implementation converts from its internal charset to that requested.

993

994 "attributes-natural-language" (naturalLanguage):

995 This operation attribute identifies the natural language used by any 'text' and 'name' attributes that
996 the IPP object is returning in this response. Unlike the "attributes-charset" operation attribute,
997 the IPP object **NEED NOT** return the same value as that supplied by the client in the request.
998 The IPP object **MAY** return the natural language of the Job object or the Printer's configured
999 natural language as identified by the Printer object's "natural-language-configured" attribute,
1000 rather than the natural language supplied by the client. For any 'text' or 'name' attribute or status
1001 message in the response that is in a different natural language than the value returned in the
1002 "attributes-natural-language" operation attribute, the IPP object **MUST** use the Natural Language
1003 Override mechanism (see sections 4.1.1.2 and 4.1.2.2) on each attribute value returned. The IPP
1004 object **MAY** use the Natural Language Override mechanism redundantly, i.e., use it even when
1005 the value is in the same natural language as the value supplied in the "attributes-natural-
1006 language" operation attribute of the response.

1007 3.1.5 Operation Targets

1008 All IPP operations are directed at IPP objects. For Printer operations, the operation is always directed at
1009 a Printer object using one of its URIs (i.e., one of the values in the Printer object's "printer-uri-
1010 supported" attribute). Even if the Printer object supports more than one URI, the client supplies only one
1011 URI as the target of the operation. The client identifies the target object by supplying the correct URI in
1012 the "printer-uri (uri)" operation attribute.

1013 For Job operations, the operation is directed at either:

- 1014 - The Job object itself using the Job object's URI. In this case, the client identifies the target object
1015 by supplying the correct URI in the "job-uri (uri)" operation attribute.
- 1016 - The Printer object that created the Job object using both the Printer objects URI and the Job object's
1017 Job ID. Since the Printer object that created the Job object generated the Job ID, it MUST be
1018 able to correctly associate the client supplied Job ID with the correct Job object. The client
1019 supplies the Printer object's URI in the "printer-uri (uri)" operation attribute and the Job object's
1020 Job ID in the "job-id (integer(1:MAX))" operation attribute.

1021

1022 If the operation is directed at the Job object directly using the Job object's URI, the client MUST NOT
1023 include the redundant "job-id" operation attribute.

1024 The operation target attributes are REQUIRED operation attributes that MUST be included in every
1025 operation request. Like the charset and natural language attributes (see section 3.1.4), the operation
1026 target attributes are specially ordered operation attributes. In all cases, the operation target attributes
1027 immediately follow the "attributes-charset" and "attributes-natural-language" attributes within the
1028 operation attribute group, however the specific ordering rules are:

- 1029 - In the case where there is only one operation target attribute (i.e., either only the "printer-uri"
1030 attribute or only the "job-uri" attribute), that attribute MUST be the third attribute in the
1031 operation attributes group.
- 1032 - In the case where Job operations use two operation target attributes (i.e., the "printer-uri" and "job-
1033 id" attributes), the "printer-uri" attribute MUST be the third attribute and the "job-id" attribute
1034 MUST be the fourth attribute.

1035

1036 In all cases, the target URIs contained within the body of IPP operation requests and responses must be
1037 in absolute format rather than relative format (a relative URL identifies a resource with the scope of the
1038 HTTP server, but does not include scheme, host or port).

1039 The following rules apply to the use of port numbers in URIs that identify IPP objects:

- 1040 1. If the URI scheme allows the port number to be explicitly included in the URI string, and a port
1041 number is specified within the URI, then that port number MUST be used by the client to contact
1042 the IPP object.
- 1043 2. If the URI scheme allows the port number to be explicitly included in the URI string, and a port
1044 number is not specified within the URI, then default port number implied by that URI scheme
1045 MUST be used by the client to contact the IPP object.

1046

1047

1048 3. If the URI scheme does not allow an explicit port number to be specified within the URI, then the
 1049 default port number implied by that URI MUST be used by the client to contact the IPP object.
 1050

1051 Note: The IPP "Encoding and Transport document [IPP-PRO] shows a mapping of IPP onto HTTP/1.1
 1052 and defines a new default port number for using IPP over HTTP/1.1.

1053 3.1.6 Operation Status Codes and Messages

1054 Every operation response includes a REQUIRED "status-code" parameter and an OPTIONAL "status-
 1055 message" operation attribute. The "status-code" provides information on the processing of a request. A
 1056 "status-message" attribute provides a short textual description of the status of the operation. The status
 1057 code is intended for use by automata, and the status message is intended for the human end user. The
 1058 "status-message" is especially useful for a later version of a Printer object to return as supplemental
 1059 information for the human user to accompany a status code that an earlier version of a client might not
 1060 understand. If a response does include a "status-message" attribute, an IPP client NEED NOT examine
 1061 or display the message, however it SHOULD do so in some implementation specific manner.

1062 The "status-code" value is a numeric value that has semantic meaning. The "status-code" syntax is
 1063 similar to a "type2 enum" (see section 4.1 on "Attribute Syntaxes") except that values can range only
 1064 from 0x0000 to 0x7FFF. Section 13 describes the status codes, assigns the numeric values, and suggests
 1065 a corresponding status message for each status code. The "status-message" attribute's syntax is
 1066 "text(255)". A client implementation of IPP SHOULD convert status code values into any localized
 1067 message that has semantic meaning to the end user.

1068 If the Printer object supports the "status-message" operation attribute, the Printer object MUST be able
 1069 to generate this message in any of the natural languages identified by the Printer object's "generated-
 1070 natural-language-supported" attribute (see the "attributes-natural-language" operation attribute specified
 1071 in section 3.1.4.1). As described in section 3.1.4.1 for any returned 'text' attribute, if there is a choice for
 1072 generating this message, the Printer object uses the natural language indicated by the value of the
 1073 "attributes-natural-language" in the client request if supported, otherwise the Printer object uses the
 1074 value in the Printer object's own "natural-language-configured" attribute. If the Printer object supports
 1075 the "status-message" operation attribute, it SHOULD use the REQUIRED 'utf-8' charset to return a status
 1076 message for the following error status codes (see section 13): 'client-error-bad-request', 'client-error-
 1077 charset-not-supported', 'server-error-internal-error', 'server-error-operation-not-supported', and 'server-
 1078 error-version-not-supported'. In this case, it MUST set the value of the "attributes-charset" operation
 1079 attribute to 'utf-8' in the error response.

1080 If the Printer performs an operation with no errors and it encounters no problems, it MUST return the
 1081 status code 'successful-ok' in the response. See section 13.

1082 If the client supplies unsupported values for the following parameters or Operation attributes, the Printer
 1083 object MUST reject the operation, NEED NOT return the unsupported attribute value in the
 1084 Unsupported Attributes group, and MUST return the indicated status code:

1085

<u>Parameter/Attribute</u>	<u>Status code</u>
----------------------------	--------------------

<u>version-number</u>	<u>server-error-version-not-supported</u>
<u>operation-id</u>	<u>server-error-operation-not-supported</u>
<u>attributes-charset</u>	<u>client-error-charset-not-supported</u>
<u>compression</u>	<u>client-error-compression-not-supported</u>
<u>document-format</u>	<u>client-error-document-format-not-supported</u>
<u>document-uri</u>	<u>client-error-uri-scheme-not-supported, client-error-document-access-error</u>

1086

1087 If the client supplies unsupported values for other attributes, or unsupported attributes, the Printer returns
 1088 the status code defined in the next section on Unsupported Attributes.

1089 3.1.7 Unsupported Attributes

1090 The Unsupported Attributes group contains attributes that are not supported by the operation. This group
 1091 is primarily for the job creation operations, but all operations can return this group.

1092

1093 A Printer object MUST include an Unsupported Attributes group in a response if the status code is one
 1094 of the following: 'successful-ok-ignored-or-substituted-attributes', 'successful-ok-conflicting-attributes',
 1095 'client-error-attributes-or-values-not-supported' or 'client-error-conflicting-attributes'.

1096

1097 If the status code is one of the four specified in the preceding paragraph, the Unsupported Attributes
 1098 group MUST contain all of those attributes and only those attributes that are:

1099

1100 a) an Operation or Job Template attribute supplied in the request, and
 1101 b) unsupported by the printer. See below for details on the three categories "unsupported"
 1102 attributes. Issue 18, Issue 23, and Issue 27

1103

1104 If the Printer object is not returning any Unsupported Attributes in the response, the Printer object
 1105 SHOULD omit Group 2 rather than sending an empty group. However, a client MUST be able to accept
 1106 an empty group.

1107

1108 Unsupported attributes fall into three categories:

1109

- 1110 1. The Printer object does not support the supplied attribute (no matter what the attribute syntax
 1111 or value).
- 1112 2. The Printer object does support the attribute, but does not support some or all of the particular
 1113 attribute syntaxes or values supplied by the client (i.e., the Printer object does not have those
 1114 attribute syntaxes or values in its corresponding "xxx-supported" attribute).
- 1115 3. The Printer object does support the attributes and values supplied, but the particular values are
 1116 in conflict with one another, because they violate a constraint, such as not being able to staple
 1117 transparencies.

1118 In the case of an unsupported attribute name, the Printer object returns the client-supplied attribute with
 1119 a substituted value of 'unsupported'. This value's syntax type is "out-of-band" and its encoding is defined
 1120 by special rules for "out-of-band" values in the "Encoding and Transport" specification [IPP-PRO]. Its
 1121 value indicates no support for the attribute itself (see the beginning of section 4.1). Issue 12

1122

1123 In the case of a supported attribute with one or more unsupported attribute syntaxes or values, the Printer
1124 object simply returns the client-supplied attribute with the unsupported attribute syntaxes or values as
1125 supplied by the client. This indicates support for the attribute, but no support for that particular attribute
1126 syntax or value. If the client supplies a multi-valued attribute with more than one value and the Printer
1127 object supports the attribute but only supports a subset of the client-supplied attribute syntaxes or values,
1128 the Printer object MUST return only those attribute syntaxes or values that are unsupported.

1129
1130 In the case of two (or more) supported attribute values that are in conflict with one another (although
1131 each is supported independently, the values conflict when requested together within the same job), the
1132 Printer object MUST return all the values that it ignores or substitutes to resolve the conflict, but not any
1133 of the values that it is still using. The choice for exactly how to resolve the conflict is implementation
1134 dependent. See sections 3.2.1.2 and 15. See The Implementer's Guide [IPP-IIG] for an example.

1135 3.1.8 Versions

1136 Each operation request and response carries with it a "version-number" parameter. Each value of the
1137 "version-number" is in the form "X.Y" where X is the major version number and Y is the minor version
1138 number. By including a version number in the client request, it allows the client to identify which
1139 version of IPP it is interested in using. If the IPP object does not support that version, the object
1140 responds with a status code of 'server-error-version-not-supported' along with the closest version number
1141 that is supported (see section 13.1.5.4).

1142 There is no version negotiation per se. However, if after receiving a 'server-error-version-not-supported'
1143 status code from an IPP object, there is nothing that prevents a client from trying again with a different
1144 version number. In order to conform to IPP/1.1, an IPP object implementations MUST support versions
1145 '1.1' ~~and SHOULD support version '1.0'~~. **Issue 33**

1146 There is only one notion of "version number" that covers both IPP Model and IPP Protocol changes.
1147 Thus the version number MUST change when introducing a new version of the Model and Semantics
1148 document [IPP-MOD] or a new version of the "Encoding and Transport" document [IPP-PRO].

1149 Changes to the major version number indicate structural or syntactic changes that make it impossible for
1150 older version of IPP clients and Printer objects to correctly parse and correctly process the new or
1151 changed attributes, operations and responses. If the major version number changes, the minor version
1152 numbers is set to zero. As an example, adding the REQUIRED "ipp-attribute-fidelity" attribute to
1153 version '1.1' (if it had not been part of version '1.1'), would have required a change to the major version
1154 number, since an IPP/1.0 Printer would not have processed a request with the correct semantics that
1155 contained the "ipp-attribute-fidelity" attribute that it did not know about. Items that might affect the
1156 changing of the major version number include any changes to the Model and Semantics document [IPP-
1157 MOD] or the "Encoding and Transport" document [IPP-PRO] itself, such as:

- 1158 - reordering of ordered attributes or attribute sets
- 1159 - changes to the syntax of existing attributes
- 1160 ~~changing Operation or Job Template attributes from OPTIONAL to REQUIRED and vice versa~~
- 1161 ~~adding REQUIRED (for an IPP object to support) operation attributes~~
- 1162 - adding REQUIRED (for an IPP object to support) operation attribute groups
- 1163 - adding values to existing REQUIRED operation attributes

1164 - adding REQUIRED operations
1165

1166 Changes to the minor version number indicate the addition of new features, attributes and attribute
1167 values that may not be understood by all IPP objects, but which can be ignored if not understood. Items
1168 that might affect the changing of the minor version number include any changes to the model objects and
1169 attributes but not the encoding and transport rules [IPP-PRO] (except adding attribute syntaxes).

1170 Examples of such changes are:

- 1171 - grouping all extensions not included in a previous version into a new version
- 1172 - adding new attribute values
- 1173 - adding new object attributes
- 1174 - adding OPTIONAL (for an IPP object to support) operation attributes (i.e., those attributes that an
1175 IPP object can ignore without confusing clients)
- 1176 - adding OPTIONAL (for an IPP object to support) operation attribute groups (i.e., those attributes
1177 that an IPP object can ignore without confusing clients)
- 1178 - adding new attribute syntaxes
- 1179 - adding OPTIONAL operations
- 1180 - changing Job Description attributes or Printer Description attributes from OPTIONAL to
1181 REQUIRED or vice versa.
- 1182 - adding OPTIONAL attribute syntaxes to an existing attribute. Issue 33
1183

1184 The encoding of the "version-number" MUST NOT change over any version number (either major or
1185 minor). This rule guarantees that all future versions will be backwards compatible with all previous
1186 versions (at least for checking the "version-number"). In addition, any protocol elements (attributes,
1187 error codes, tags, etc.) that are not carried forward from one version to the next are deprecated so that
1188 they can never be reused with new semantics.

1189 Implementations that support a certain ~~major~~-version NEED NOT support ALL previous versions. As
1190 each new ~~major~~-version is defined (through the release of a new specification), that major version will
1191 specify which previous ~~major~~-versions MUST and which versions SHOULD be supported in compliant
1192 implementations. Issue 33

1193 3.1.9 Job Creation Operations

1194 In order to "submit a print job" and create a new Job object, a client issues a create request. A create
1195 request is any one of following three operation requests:

- 1196 - The Print-Job Request: A client that wants to submit a print job with only a single document uses
1197 the Print-Job operation. The operation allows for the client to "push" the document data to the
1198 Printer object by including the document data in the request itself.
1199
- 1200 - The Print-URI Request: A client that wants to submit a print job with only a single document
1201 (where the Printer object "pulls" the document data instead of the client "pushing" the data to the
1202 Printer object) uses the Print-URI operation. In this case, the client includes in the request only a
1203 URI reference to the document data (not the document data itself).
1204

1205 - The Create-Job Request: A client that wants to submit a print job with multiple documents uses the
1206 Create-Job operation. This operation is followed by an arbitrary number of Send-Document
1207 and/or Send-URI operations (each creating another document for the newly create Job object).
1208 The Send-Document operation includes the document data in the request (the client "pushes" the
1209 document data to the printer), and the Send-URI operation includes only a URI reference to the
1210 document data in the request (the Printer "pulls" the document data from the referenced location).
1211 The last Send-Document or Send-URI request for a given Job object includes a "last-document"
1212 operation attribute set to 'true' indicating that this is the last request.
1213

1214 Throughout this model specification, the term "create request" is used to refer to any of these three
1215 operation requests.

1216 A Create-Job operation followed by only one Send-Document operation is semantically equivalent to a
1217 Print-Job operation, however, for performance reasons, the client SHOULD use the Print-Job operation
1218 for all single document jobs. Also, Print-Job is a REQUIRED operation (all implementations MUST
1219 support it) whereas Create-Job is an OPTIONAL operation, hence some implementations might not
1220 support it.

1221 Job submission time is the point in time when a client issues a create request. The initial state of every
1222 Job object is the 'pending', ~~or~~ 'pending-held', or 'processing' state (see section 4.3.7). **Issue 13** Later,
1223 **When** the Printer object begins processing the print job. ~~At this point in time~~, the Job object's state
1224 moves to 'processing'. This is known as job processing time. There are validation checks that must be
1225 done at job submission time and others that must be performed at job processing time.

1226 At job submission time and at the time a Validate-Job operation is received, the Printer MUST do the
1227 following:

- 1228 1. Process the client supplied attributes and either accept or reject the request
 - 1229 2. Validate the syntax of and support for the scheme of any client supplied URI
- 1230

1231 At job submission time the Printer object MUST validate whether or not the supplied attributes, attribute
1232 syntaxes, and values are supported by matching them with the Printer object's corresponding "xxx-
1233 supported" attributes. See section 3.1.7~~3.2.1.2~~ for details. [IPP-IIG] presents suggested steps for an IPP
1234 object to either accept or reject any request and additional steps for processing create requests.

1235 At job submission time the Printer object NEED NOT perform the validation checks reserved for job
1236 processing time such as:

- 1237 1. Validating the document data
 - 1238 2. Validating the actual contents of any client supplied URI (resolve the reference and follow the link
1239 to the document data)
- 1240

1241 At job submission time, these additional job processing time validation checks are essentially useless,
1242 since they require actually parsing and interpreting the document data, are not guaranteed to be 100%
1243 accurate, and MUST be done, yet again, at job processing time. Also, in the case of a URI, checking for
1244 availability at job submission time does not guarantee availability at job processing time. In addition, at

1245 job processing time, the Printer object might discover any of the following conditions that were not
1246 detectable at job submission time:

- 1247 - runtime errors in the document data,
- 1248 - nested document data that is in an unsupported format,
- 1249 - the URI reference is no longer valid (i.e., the server hosting the document might be down), or
- 1250 - any other job processing error

1251

1252 At job submission time, a Printer object, especially a non-spooling Printer, MAY accept jobs that it does
1253 not have enough space for. In such a situation, a Printer object MAY stop reading data from a client for
1254 an indefinite period of time. A client MUST be prepared for a write operation to block for an indefinite
1255 period of time (See section 5.1 on client conformance).

1256 When a Printer object has too little space for starting a new job, it MAY reject a new create request. In
1257 this case, a Printer object MUST return a response (in reply to the rejected request) with a status-code of
1258 'server-error-busy' (See section 14.1.5.8) and it MAY close the connection before receiving all bytes of
1259 the operation. When receiving a 'server-error-busy' status-code in an operation response, a client MUST
1260 be prepared for the Printer object to close the connection before the client has sent all of the data
1261 (especially for the Print-Job operation). A client MUST be prepared to keep submitting a create request
1262 until the IPP Printer object accepts the create request. Issue 20

1263 At job processing time, since the Printer object has already responded with a successful status code in
1264 the response to the create request, if the Printer object detects an error, the Printer object is unable to
1265 inform the end user of the error with an operation status code. In this case, the Printer, depending on the
1266 error, can set the **job object's** "job-state", "job-state-reasons", or "job-state-message" attributes to the
1267 appropriate value(s) so that later queries can report the correct job status.

1268 Note: Asynchronous notification of events is outside the scope of this IPP/1.1 document.

1269 3.2 Printer Operations

1270 All Printer operations are directed at Printer objects. A client MUST always supply the "printer-uri"
1271 operation attribute in order to identify the correct target of the operation.

1272 3.2.1 Print-Job Operation

1273 This REQUIRED operation allows a client to submit a print job with only one document and supply the
1274 document data (rather than just a reference to the data). See Section 15 for the suggested steps for
1275 processing create operations and their Operation and Job Template attributes.

1276 3.2.1.1 Print-Job Request

1277 The following groups of attributes are supplied as part of the Print-Job Request:

1278 Group 1: Operation Attributes

1279 Natural Language and Character Set:

1280 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1281 3.1.4.1. The Printer object MUST copy these values to the corresponding Job Description
1282 attributes described in sections 4.3.17 and 4.3.18.

1283

1284 Target:

1285 The "printer-uri" (uri) operation attribute which is the target for this operation as described in
1286 section 3.1.5.

1287

1288 Requesting User Name:

1289 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
1290 described in section 8.3.

1291

1292 "job-name" (name(MAX)):

1293 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1294 attribute. It contains the client supplied Job name. If this attribute is supplied by the client, its
1295 value is used for the "job-name" attribute of the newly created Job object. The client MAY
1296 automatically include any information that will help the end-user distinguish amongst his/her
1297 jobs, such as the name of the application program along with information from the document,
1298 such as the document name, document subject, or source file name. If this attribute is not
1299 supplied by the client, the Printer generates a name to use in the "job-name" attribute of the
1300 newly created Job object (see Section 4.3.5).

1301

1302 "ipp-attribute-fidelity" (boolean):

1303 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1304 attribute. The value 'true' indicates that total fidelity to client supplied Job Template attributes
1305 and values is required, else the Printer object MUST reject the Print-Job request. The value
1306 'false' indicates that a reasonable attempt to print the Job object is acceptable and the Printer
1307 object MUST accept the Print-job request. If not supplied, the Printer object assumes the value is
1308 'false'. All Printer objects MUST support both types of job processing. See section 15 for a full
1309 description of "ipp-attribute-fidelity" and its relationship to other attributes, especially the Printer
1310 object's "pdl-override-supported" attribute.

1311

1312 "document-name" (name(MAX)):

1313 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1314 attribute. It contains the client supplied document name. The document name MAY be
1315 different than the Job name. Typically, the client software automatically supplies the document
1316 name on behalf of the end user by using a file name or an application generated name. If this
1317 attribute is supplied, its value can be used in a manner defined by each implementation.
1318 Examples include: printed along with the Job (job start sheet, page adornments, etc.), used by
1319 accounting or resource tracking management tools, or even stored along with the document as a
1320 document level attribute. IPP/1.1 does not support the concept of document level attributes.

1321

1322 "compression" (type3 keyword)

1323 The client OPTIONALLY supplies this attribute. The Printer object ~~OPTIONALLY~~**MUST**
1324 supports this attribute and the "compression-supported" attribute (see section 4.4.32). The client
1325 supplied "compression" operation attribute identifies the compression algorithm used on the
1326 document data. The following cases exist:

- 1327 a) -If the client omits this attribute, the Printer object MUST assume that the data is not
1328 compressed (i.e. the Printer follows the rules below as if the client supplied the
1329 "compression" attribute with a value of 'none').
- 1330 b) ~~If the client supplies the attribute and the Printer object supports the attribute, the~~
1331 ~~Printer object uses the corresponding decompression algorithm on the document data.~~
1332 If the client supplies this attribute, but the value is not supported by the Printer object,
1333 i.e., the value is not one of the values of the Printer object's "compression-supported"
1334 attribute, the Printer object MUST ~~copy the attribute and its value to the Unsupported~~
1335 ~~Attributes response group,~~ reject the request, and return the 'client-error-attributes-or-
1336 valuescompression-not-supported' status code. ~~If the client supplies this attribute, but~~
1337 ~~this attribute is not supported by the Printer object, i.e., the "compression-supported"~~
1338 ~~attribute is not one of the Printer's Printer Description attributes, the Printer object~~
1339 ~~MUST copy the attribute to the Unsupported Attributes response group changing the~~
1340 ~~value to the out-of-band 'unsupported' value (see section 4.1), reject the request, and~~
1341 ~~return the 'client-error-attributes-or-values-not-supported' status code.~~ See section
1342 3.1.73.2.1.2 for returning unsupported attributes and values.
- 1343 c) ~~If the client supplies the attribute and the Printer object supports the attribute value,~~
1344 ~~the Printer object uses the corresponding decompression algorithm on the document~~
1345 ~~data.~~
- 1346 d) ~~If the decompression algorithm fails before the Printer sends~~returns an operation
1347 ~~response, the Printer object must~~**MUST** reject the request and return the 'client-error-
1348 ~~compression-error' status code.~~
- 1349 e) ~~If the decompression algorithm fails after the Printer sends~~returns an operation
1350 ~~response-, the Printer object must~~**MUST** abort the job and add the 'compression-error'
1351 ~~value to the job's "job-state-reasons" attribute.~~
- 1352 f) ~~If the decompression algorithm succeeds, the document data MUST then have the~~
1353 ~~format specified by the job's "document-format" attribute (q.v.).~~ Issue 28

1354
1355 "document-format" (mimeType) :

1356 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1357 attribute. The value of this attribute identifies the format of the supplied document data. The
1358 following cases exist:

- 1359 a) If the client does not supply this attribute, the Printer object assumes that the
1360 document data is in the format defined by the Printer object's "document-format-
1361 default" attribute. (i.e. the Printer follows the rules below as if the client supplied the
1362 "document-format" attribute with a value equal to the printer's default value).
- 1363 b) If the client supplies this attribute, but the value is not supported by the Printer object,
1364 i.e., the value is not one of the values of the Printer object's "document-format-
1365 supported" attribute, the Printer object MUST ~~(if specified in the job request)~~ reject

1366 the request and return the 'client-error-document-format-not-supported' status
 1367 code ~~onestatus code~~.

- 1368 c) If the client supplies this attribute and its value is 'application/octet-stream' (i.e. to be
 1369 auto-sensed, see Section 4.1.914.1.9.1), and the format is not one of the document-
 1370 formats that the Printer can auto-sense, and this check occurs before the Printer
 1371 sends returns an operation response, then the Printer MUST reject the request and
 1372 return the 'client-error-document-format-not-supported' status code.
- 1373 d) If the client supplies this attribute, and the value is supported by the Printer object, the
 1374 document data, the Printer is capable of interpreting the document data.
- 1375 e) If interpreting of the document data fails before the Printer sends returns an operation
 1376 response, the Printer object must MUST reject the request and return the 'client-error-
 1377 document-format-error' status code.
- 1378 f) If interpreting of the document data fails after the Printer sends returns an operation
 1379 response, the Printer object must MUST abort the job and add the 'document-format-
 1380 error' value to the job's "job-state-reasons" attribute. Issue 11

1381
 1382 "document-natural-language" (naturalLanguage):

1383 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports
 1384 this attribute. This attribute specifies the natural language of the document for those document-
 1385 formats that require a specification of the natural language in order to image the document
 1386 unambiguously. There are no particular values required for the Printer object to support.

1387
 1388
 1389 "job-k-octets" (integer(0:MAX))

1390 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports
 1391 this attribute and the "job-k-octets-supported" attribute (see section 4.4.33). The client supplied
 1392 "job-k-octets" operation attribute identifies the total size of the document(s) in K octets being
 1393 submitted (see section 4.3.15.1 for the complete semantics). If the client supplies the attribute
 1394 and the Printer object supports the attribute, the value of the attribute is used to populate the Job
 1395 object's "job-k-octets" Job Description attribute.

1396
 1397 Note: For this attribute and the following two attributes ("job-impressions", and "job-media-
 1398 sheets"), if the client supplies the attribute, but the Printer object does not support the attribute,
 1399 the Printer object ignores the client-supplied value. If the client supplies the attribute and the
 1400 Printer supports the attribute, and the value is within the range of the corresponding Printer
 1401 object's "xxx-supported" attribute, the Printer object MUST use the value to populate the Job
 1402 object's "xxx" attribute. If the client supplies the attribute and the Printer supports the attribute,
 1403 but the value is outside the range of the corresponding Printer object's "xxx-supported" attribute,
 1404 the Printer object MUST copy the attribute and its value to the Unsupported Attributes response
 1405 group, reject the request, and return the 'client-error-attributes-or-values-not-supported' status
 1406 code. If the client does not supply the attribute, the Printer object MAY choose to populate the
 1407 corresponding Job object attribute depending on whether the Printer object supports the attribute
 1408 and is able to calculate or discern the correct value.
 1409

1410 "job-impressions" (integer(0:MAX))

1411 The client **OPTIONALLY** supplies this attribute. The Printer object **OPTIONALLY** supports
1412 this attribute and the "job-impressions-supported" attribute (see section 4.4.34). The client
1413 supplied "job-impressions" operation attribute identifies the total size in number of impressions
1414 of the document(s) being submitted (see section 4.3.15.2 for the complete semantics).

1415
1416 See note under "job-k-octets".

1417
1418 "job-media-sheets" (integer(0:MAX))

1419 The client **OPTIONALLY** supplies this attribute. The Printer object **OPTIONALLY** supports
1420 this attribute and the "job-media-sheets-supported" attribute (see section 4.4.35). The client
1421 supplied "job-media-sheets" operation attribute identifies the total number of media sheets to be
1422 produced for this job (see section 4.3.15.3 for the complete semantics).

1423
1424 See note under "job-k-octets".

1425

1426 Group 2: Job Template Attributes

1427 The client **OPTIONALLY** supplies a set of Job Template attributes as defined in section 4.2. If
1428 the client is not supplying any Job Template attributes in the request, the client **SHOULD** omit
1429 Group 2 rather than sending an empty group. However, a Printer object **MUST** be able to accept
1430 an empty group.

1431

1432 Group 3: Document Content

1433 The client **MUST** supply the document data to be processed.

1434

1435 Note: In addition to the **MANDATORY** parameters required for every operation request, the simplest
1436 Print-Job Request consists of just the "attributes-charset" and "attributes-natural-language" operation
1437 attributes; the "printer-uri" target operation attribute; the Document Content and nothing else. In this
1438 simple case, the Printer object:

- 1439 - creates a new Job object (the Job object contains a single document),
- 1440 - stores a generated Job name in the "job-name" attribute in the natural language and charset
1441 requested (see Section 3.1.4.1) (if those are supported, otherwise using the Printer object's default
1442 natural language and charset), and
- 1443 - at job processing time, uses its corresponding default value attributes for the supported Job
1444 Template attributes that were not supplied by the client as IPP attribute or embedded instructions
1445 in the document data.

1446

1447 3.2.1.2 Print-Job Response

1448 The Printer object **MUST** return to the client the following sets of attributes as part of the Print-Job
1449 Response:

1450 Group 1: Operation Attributes

1451 Status Message:

1452 In addition to the REQUIRED status code returned in every response, the response
1453 OPTIONALLY includes a "status-message" (text) operation attribute as described in sections 13
1454 and 3.1.6. If the client supplies unsupported or conflicting Job Template attributes or values, the
1455 Printer object MUST reject or accept the Print-Job request depending on the whether the client
1456 supplied a 'true' or 'false' value for the "ipp-attribute-fidelity" operation attribute. See the
1457 Implementer's Guide [IPP-IIG] for a complete description of the suggested steps for processing a
1458 create request.

1459

1460 Natural Language and Character Set:

1461 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1462 3.1.4.2.

1463

1464 Group 2: Unsupported Attributes

1465 [See section 3.1.7 for details on returning Unsupported Attributes.](#)

1466 ~~This is a set of Operation and Job Template attributes supplied by the client (in the request) that~~
1467 ~~are not supported by the Printer object or that conflict with one another (see the Implementer's~~
1468 ~~Guide [IPP-IIG]).~~

1469

1470 ~~In these three cases,~~ The value of the "ipp-attribute-fidelity" supplied by the client does not
1471 affect what [attributes](#) the Printer object returns [in this group](#). The value of "ipp-attribute-fidelity"
1472 only affects whether the Print-Job operation is accepted or rejected. If the job is accepted, the
1473 client may query the job using the Get-Job-Attributes operation requesting the unsupported
1474 attributes that were returned in the create response to see which attributes were ignored (not
1475 stored on the Job object) and which attributes were stored with other (substituted) values.

1476

1477 Group 3: Job Object Attributes

1478 "job-uri" (uri):

1479 The Printer object MUST return the Job object's URI by returning the contents of the
1480 REQUIRED "job-uri" Job object attribute. The client uses the Job object's URI when directing
1481 operations at the Job object. The Printer object always uses its configured security policy when
1482 creating the new URI. However, if the Printer object supports more than one URI, the Printer
1483 object also uses information about which URI was used in the Print-Job Request to generated the
1484 new URI so that the new URI references the correct access channel. In other words, if the Print-
1485 Job Request comes in over a secure channel, the Printer object MUST generate a Job URI that
1486 uses the secure channel as well.

1487

1488 "job-id" (integer(1:MAX)):

1489 The Printer object MUST return the Job object's Job ID by returning the REQUIRED "job-id"
1490 Job object attribute. The client uses this "job-id" attribute in conjunction with the "printer-uri"
1491 attribute used in the Print-Job Request when directing Job operations at the Printer object.

1492

1493 "job-state":

1494 The Printer object MUST return the Job object's REQUIRED "job-state" attribute. The value of
1495 this attribute (along with the value of the next attribute: "job-state-reasons") is taken from a
1496 "snapshot" of the new Job object at some meaningful point in time (implementation defined)
1497 between when the Printer object receives the Print-Job Request and when the Printer object
1498 returns the response.

1499
1500 "job-state-reasons":

1501 The Printer object ~~OPTIONALLY~~MUST returns the Job object's ~~OPTIONAL~~REQUIRED "job-
1502 state-reasons" attribute. ~~If the Printer object supports this attribute then it MUST be returned in~~
1503 ~~the response. If this attribute is not returned in the response, the client can assume that the "job-~~
1504 ~~state-reasons" attribute is not supported and will not be returned in a subsequent Job object~~
1505 ~~query.~~ **Issue 30**

1506
1507 "job-state-message":

1508 The Printer object OPTIONALLY returns the Job object's OPTIONAL "job-state-message"
1509 attribute. If the Printer object supports this attribute then it MUST be returned in the response. If
1510 this attribute is not returned in the response, the client can assume that the "job-state-message"
1511 attribute is not supported and will not be returned in a subsequent Job object query.

1512
1513 "number-of-intervening-jobs":

1514 The Printer object OPTIONALLY returns the Job object's OPTIONAL "number-of-intervening-
1515 jobs" attribute. If the Printer object supports this attribute then it MUST be returned in the
1516 response. If this attribute is not returned in the response, the client can assume that the "number-
1517 of-intervening-jobs" attribute is not supported and will not be returned in a subsequent Job object
1518 query.

1519
1520 Note: Since any printer state information which affects a job's state is reflected in the "job-state"
1521 and "job-state-reasons" attributes, it is sufficient to return only these attributes and no specific
1522 printer status attributes.

1523
1524 Note: In addition to the MANDATORY parameters required for every operation response, the simplest
1525 response consists of the just the "attributes-charset" and "attributes-natural-language" operation
1526 attributes and the "job-uri", "job-id", and "job-state" Job Object Attributes. In this simplest case, the
1527 status code is 'successful-ok' ~~"successful-ok"~~ and there is no "status-message" operation attribute.

1528 3.2.2 Print-URI Operation

1529 This OPTIONAL operation is identical to the Print-Job operation (section 3.2.1) except that a client
1530 supplies a URI reference to the document data using the "document-uri" (uri) operation attribute (in
1531 Group 1) rather than including the document data itself. Before returning the response, the Printer
1532 MUST validate that the Printer supports the retrieval method (e.g., http, ftp, etc.) implied by the URI,
1533 and MUST check for valid URI syntax. If the client-supplied URI scheme is not supported, i.e. the value
1534 is not in the Printer object's "referenced-uri-scheme-supported" attribute, the Printer object MUST reject
1535 the request and return the 'client-error-uri-scheme-not-supported' status code.

1536 The IPP Printer MAY validate the accessibility of the document as part of the operation or subsequently.
1537 If the Printer determines an accessibility problem before returning an operation response, it rejects the
1538 request and returns the 'client-error-document-access-error' status code. If the Printer determines this
1539 accessibility problem after accepting the request and returning an operation response with one of the
1540 successful status codes, the Printer adds the 'document-access-error' value to the job's "job-state-reasons"
1541 attribute. See The Implementer's Guide [IPP-IIG] for suggested additional checks. The Printer NEED
1542 NOT follow the reference and validate the contents of the reference. Issue 35

1543 If the Printer object supports this operation, it MUST support the "reference-uri-schemes-supported"
1544 Printer attribute (see section 4.4.27).

1545 It is up to the IPP object to interpret the URI and subsequently "pull" the document from the source
1546 referenced by the URI string.

1547 3.2.3 Validate-Job Operation

1548 This REQUIRED operation is similar to the Print-Job operation (section 3.2.1) except that a client
1549 supplies no document data and the Printer allocates no resources (i.e., it does not create a new Job
1550 object). This operation is used only to verify capabilities of a printer object against whatever attributes
1551 are supplied by the client in the Validate-Job request. By using the Validate-Job operation a client can
1552 validate that an identical Print-Job operation (with the document data) would be accepted. The Validate-
1553 Job operation also performs the same security negotiation as the Print-Job operation (see section 8), so
1554 that a client can check that the client and Printer object security requirements can be met before
1555 performing a Print-Job operation.

1556 Note: The Validate-Job operation does not accept a "document-uri" attribute in order to allow a client to
1557 check that the same Print-URI operation will be accepted, since the client doesn't send the data with the
1558 Print-URI operation. The client SHOULD just issue the Print-URI request.

1559 The Printer object returns the same status codes, Operation Attributes (Group 1) and Unsupported
1560 Attributes (Group 2) as the Print-Job operation. However, no Job Object Attributes (Group 3) are
1561 returned, since no Job object is created.

1562 3.2.4 Create-Job Operation

1563 This OPTIONAL operation is similar to the Print-Job operation (section 3.2.1) except that in the Create-
1564 Job request, a client does not supply document data or any reference to document data. Also, the client
1565 does not supply any of the "document-name", "document-format", "compression", or "document-natural-
1566 language" operation attributes. This operation is followed by one or more Send-Document or Send-URI
1567 operations. In each of those operation requests, the client OPTIONALLY supplies the "document-
1568 name", "document-format", and "document-natural-language" attributes for each document in the multi-
1569 document Job object.

1570 If a Printer object supports the Create-Job operation, it MUST also support the Send-Document
1571 operation and also MAY support the Send-URI operation.

1572 If the Printer object supports this operation, it MUST support the "multiple-operation-time-out" Printer
1573 attribute (see section 4.4.31).

1574 If the Printer object supports this operation, then it MUST support the "multiple-document-jobs-
1575 supported" Printer Description attribute and indicate whether or not it supports multiple-document jobs.
1576 **Issue 34**

1577 If the Printer object supports this operation and supports multiple documents in a job, then it MUST
1578 support the "multiple-document-handling" Job Template job attribute with at least one value (see section
1579 4.2.4) and the associated "multiple-document-handling-default" and "multiple-document-handling-
1580 supported" Job Template Printer attributes. **Issue 34**

1581 After the Create-Job operation has completed, the value of the "job-state" attribute is similar to the "job-
1582 state" after a Print-Job, even though no document-data has arrived. A Printer MAY set the 'job-data-
1583 insufficient' value of the job's "job-state-reason" attribute to indicate that processing cannot begin until
1584 sufficient data has arrived and set the "job-state" to either 'pending' or 'pending-held'. A non-spooling
1585 printer that doesn't implement the 'pending' job state may even set the "job-state" to 'processing', even
1586 though there is not yet any data to process. **Issue 13**

1587 3.2.5 Get-Printer-Attributes Operation

1588 This REQUIRED operation allows a client to request the values of the attributes of a Printer object. In
1589 the request, the client supplies the set of Printer attribute names and/or attribute group names in which
1590 the requester is interested. In the response, the Printer object returns a corresponding attribute set with
1591 the appropriate attribute values filled in.

1592 For Printer objects, the possible names of attribute groups are:

- 1593 - 'job-template': all the subset of the Job Template attributes that apply to a Printer object (the last
1594 two columns of the table in Section 4.2) that the implementation supports for Printer objects.
- 1595 - 'printer-description': the subset of the attributes specified in Section 4.4 that the implementation
1596 supports for Printer objects.
- 1597 - 'all': the special group 'all' that includes all supported-attributes that the implementation supports for
1598 Printer objects. **Issue 23**

1600 Since a client MAY request specific attributes or named groups, there is a potential that there is some
1601 overlap. For example, if a client requests, 'printer-name' and 'all', the client is actually requesting the
1602 "printer-name" attribute twice: once by naming it explicitly, and once by inclusion in the 'all' group. In
1603 such cases, the Printer object NEED NOT return each attribute only once in the response even if it is
1604 requested multiple times. The client SHOULD NOT request the same attribute in multiple ways.

1605 It is NOT REQUIRED that a Printer object support all attributes belonging to a group (since some
1606 attributes are OPTIONAL). However, it is REQUIRED that each Printer object support all group names.

1607 3.2.5.1 Get-Printer-Attributes Request

1608 The following sets of attributes are part of the Get-Printer-Attributes Request:

1609 Group 1: Operation Attributes

1610 Natural Language and Character Set:

1611 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1612 3.1.4.1.

1613

1614 Target:

1615 The "printer-uri" (uri) operation attribute which is the target for this operation as described in
1616 section 3.1.5.

1617

1618 Requesting User Name:

1619 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
1620 described in section 8.3.

1621

1622 "requested-attributes" (1setOf keyword) :

1623 The client OPTIONALLY supplies a set of attribute names and/or attribute group names in
1624 whose values the requester is interested. The Printer object MUST support this attribute. If the
1625 client omits this attribute, the Printer MUST respond as if this attribute had been supplied with a
1626 value of 'all'.

1627

1628 "document-format" (mimeMediaType) :

1629 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1630 attribute. This attribute is useful for a Printer object to determine the set of supported attribute
1631 values that relate to the requested document format. The Printer object MUST return the
1632 attributes and values that it uses to validate a job on a create or Validate-Job operation in which
1633 this document format is supplied. The Printer object SHOULD return only (1) those attributes
1634 that are supported for the specified format and (2) the attribute values that are supported for the
1635 specified document format. By specifying the document format, the client can get the Printer
1636 object to eliminate the attributes and values that are not supported for a specific document
1637 format. For example, a Printer object might have multiple interpreters to support both
1638 'application/postscript' (for PostScript) and 'text/plain' (for text) documents. However, for only
1639 one of those interpreters might the Printer object be able to support "number-up" with values of
1640 '1', '2', and '4'. For the other interpreter it might be able to only support "number-up" with a value
1641 of '1'. Thus a client can use the Get-Printer-Attributes operation to obtain the attributes and
1642 values that will be used to accept/reject a create job operation.

1643

1644 If the Printer object does not distinguish between different sets of supported values for each
1645 different document format when validating jobs in the create and Validate-Job operations, it
1646 MUST NOT distinguish between different document formats in the Get-Printer-Attributes
1647 operation. If the Printer object does distinguish between different sets of supported values for
1648 each different document format specified by the client, this specialization applies only to the
1649 following Printer object attributes:

- 1650
1651 - Printer attributes that are Job Template attributes ("xxx-default" "xxx-supported", and "xxx-
1652 ready" in the Table in Section 4.2),
1653 - "pdl-override-supported",
1654 - "compression-supported",
1655 - "job-k-octets-supported",
1656 - "job-impressions-supported",
1657 - "job-media-sheets-supported"
1658 - "printer-driver-installer",
1659 - "color-supported", and
1660 - "reference-uri-schemes-supported"

1661
1662 The values of all other Printer object attributes (including "document-format-supported") remain
1663 invariant with respect to the client supplied document format (except for new Printer description
1664 attribute as registered according to section 6.2).

1665
1666 If the client omits this "document-format" operation attribute, the Printer object MUST respond
1667 as if the attribute had been supplied with the value of the Printer object's "document-format-
1668 default" attribute. It is recommended that the client always supply a value for "document-
1669 format", since the Printer object's "document-format-default" may be 'application/octet-stream',
1670 in which case the returned attributes and values are for the union of the document formats that
1671 the Printer can automatically sense. For more details, see the description of the
1672 'mimeType' attribute syntax in section 4.1.9.

1673
1674 If the client supplies a value for the "document-format" Operation attribute that is not supported
1675 by the Printer, i.e., is not among the values of the Printer object's "document-format-supported"
1676 attribute, the Printer object MUST reject the operation and return the 'client-error-document-
1677 format-not-supported' status code.
1678

1679 3.2.5.2 Get-Printer-Attributes Response

1680 The Printer object returns the following sets of attributes as part of the Get-Printer-Attributes Response:

1681 Group 1: Operation Attributes

1682 Status Message:

1683 In addition to the REQUIRED status code returned in every response, the response
1684 OPTIONALLY includes a "status-message" (text) operation attribute as described in section 13
1685 [and](#) 3.1.6.

1686 Natural Language and Character Set:

1687 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1688 3.1.4.2.
1689

1691 Group 2: Unsupported Attributes

1692 [See section 3.1.7 for details on returning Unsupported Attributes.](#)

1693

1694 ~~This is a set of Operation attributes supplied by the client (in the request) that are not supported~~
1695 ~~by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16).~~ The response
1696 NEED NOT contain the "requested-attributes" operation attribute with any supplied values
1697 (attribute keywords) that were requested by the client but are not supported by the IPP object. **If**
1698 **the Printer does include unsupported attributes referenced in "requested-attributes" and such**
1699 **attributes include group names, such as 'all', the unsupported attributes MUST NOT include**
1700 **attributes described in the standard but not supported by the implementation.** **Issue 23** ~~If the~~
1701 ~~Printer object is not returning any Unsupported Attributes in the response, the Printer object~~
1702 ~~SHOULD omit Group 2 rather than sending an empty group. However, a client MUST be able~~
1703 ~~to accept an empty group.~~
1704

1704

1705 Group 3: Printer Object Attributes

1706 This is the set of requested attributes and their current values. The Printer object ignores (does
1707 not respond with) any requested attribute which is not supported. The Printer object MAY
1708 respond with a subset of the supported attributes and values, depending on the security policy in
1709 force. However, the Printer object MUST respond with the 'unknown' value for any supported
1710 attribute (including all REQUIRED attributes) for which the Printer object does not know the
1711 value. Also the Printer object MUST respond with the 'no-value' for any supported attribute
1712 (including all REQUIRED attributes) for which the system administrator has not configured a
1713 value. See the description of the "out-of-band" values in the beginning of Section 4.1.
1714

1715 3.2.6 Get-Jobs Operation

1716 This REQUIRED operation allows a client to retrieve the list of Job objects belonging to the target
1717 Printer object. The client may also supply a list of Job attribute names and/or attribute group names. A
1718 group of Job object attributes will be returned for each Job object that is returned.

1719 This operation is similar to the Get-Job-Attributes operation, except that this Get-Jobs operation returns
1720 attributes from possibly more than one object (see the description of Job attribute group names in section
1721 3.3.4).

1722 3.2.6.1 Get-Jobs Request

1723 The client submits the Get-Jobs request to a Printer object.

1724 The following groups of attributes are part of the Get-Jobs Request:

1725 Group 1: Operation Attributes

1726 Natural Language and Character Set:

1727 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1728 3.1.4.1.

1729

1730 Target:

1731 The "printer-uri" (uri) operation attribute which is the target for this operation as described in
1732 section 3.1.5.

1733

1734 Requesting User Name:

1735 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
1736 described in section 8.3.

1737

1738 "limit" (integer(1:MAX)):

1739 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1740 attribute. It is an integer value that ~~indicates a limit to the number of Job objects returned.~~
1741 determines the maximum number of jobs that a client will receive from the Printer even if
1742 "which-jobs" or "my-jobs" restrict/constrain which jobs are returned. -The limit is a "stateless
1743 limit" in that if the value supplied by the client is 'N', then only the first 'N' jobs are returned in
1744 the Get-Jobs Response. There is no mechanism to allow for the next 'M' jobs after the first 'N'
1745 jobs. If the client does not supply this attribute, the Printer object responds with all applicable
1746 jobs. **Issue 8**

1747

1748 "requested-attributes" (1setOf keyword):

1749 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1750 attribute. It is a set of Job attribute names and/or attribute groups names in whose values the
1751 requester is interested. This set of attributes is returned for each Job object that is returned. The
1752 allowed attribute group names are the same as those defined in the Get-Job-Attributes operation
1753 in section 3.3.4. If the client does not supply this attribute, the Printer MUST respond as if the
1754 client had supplied this attribute with two values: 'job-uri' and 'job-id'.

1755

1756 "which-jobs" (keyword):

1757 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1758 attribute. It indicates which Job objects MUST be returned by the Printer object. The values for
1759 this attribute are:

1760

1761 'completed': This includes any Job object whose state is 'completed', 'canceled', or 'aborted'.

1762 'not-completed': This includes any Job object whose state is 'pending', 'processing',

1763 'processing-stopped', or 'pending-held'.

1764

1765 A Printer object MUST support both values. However, if the implementation does not keep jobs
1766 in the 'completed', 'canceled', and 'aborted' states, then it returns no jobs when the 'completed'
1767 value is supplied.

1768

1769 If a client supplies some other value, the Printer object MUST copy the attribute and the
1770 unsupported value to the Unsupported Attributes response group, reject the request, and return
1771 the 'client-error-attributes-or-values-not-supported' status code.

1772

1773 If the client does not supply this attribute, the Printer object MUST respond as if the client had
1774 supplied the attribute with a value of 'not-completed'.

1775

1776 "my-jobs" (boolean):

1777 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1778 attribute. It indicates whether ~~all-jobs~~ from all users or just the jobs submitted by the requesting
1779 user of this request -MUST be returned by the Printer object. If the client does not supply this
1780 attribute, the Printer object MUST respond as if the client had supplied the attribute with a value
1781 of 'false', i.e., ~~all-jobs~~ from all users. The means for authenticating the requesting user and
1782 matching the jobs is described in section 8.

1783 3.2.6.2 Get-Jobs Response

1784 The Printer object returns all of the Job objects up to the number specified by the "limit" attribute that
1785 match the criteria as defined by the attribute values supplied by the client in the request. It is possible
1786 that no Job objects are returned since there may literally be no Job objects at the Printer, or there may be
1787 no Job objects that match the criteria supplied by the client. If the client requests any Job attributes at
1788 all, there is a set of Job Object Attributes returned for each Job object.

1789 It is not an error for the Printer to return 0 jobs. If the response returns 0 jobs because there are no jobs
1790 matching the criteria, and the request would have returned 1 or more jobs with a status code of
1791 'successful-ok' if there had been jobs matching the criteria, then the status code for 0 jobs MUST be
1792 'successful-ok'. Issue 24

1793 Group 1: Operation Attributes

1794 Status Message:

1795 In addition to the REQUIRED status code returned in every response, the response
1796 OPTIONALLY includes a "status-message" (text) operation attribute as described in sections 13
1797 and 3.1.6.

1798

1799 Natural Language and Character Set:

1800 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1801 3.1.4.2.

1802

1803 Group 2: Unsupported Attributes

1804 See section 3.1.7 for details on returning Unsupported Attributes.

1805

1806 ~~This is a set of Operation attributes supplied by the client (in the request) that are not supported~~
1807 ~~by the Printer object or that conflict with one another (see sections 3.2.1.2 and the Implementer's~~
1808 ~~Guide [IPP IIG]).~~ The response NEED NOT contain the "requested-attributes" operation
1809 attribute with any supplied values (attribute keywords) that were requested by the client but are
1810 not supported by the IPP object. If the Printer does include unsupported attributes referenced in
1811 "requested-attributes" and such attributes include group names, such as 'all', the unsupported
1812 attributes MUST NOT include attributes described in the standard but not supported by the
1813 implementation. Issue 23 ~~If the Printer object is not returning any Unsupported Attributes in the~~
1814 ~~response, the Printer object SHOULD omit Group 2 rather than sending an empty group.~~
1815 ~~However, a client MUST be able to accept an empty group.~~

1816

1817 Groups 3 to N: Job Object Attributes

1818 The Printer object responds with one set of Job Object Attributes for each returned Job object.
1819 The Printer object ignores (does not respond with) any requested attribute or value which is not
1820 supported or which is restricted by the security policy in force, including whether the requesting
1821 user is the user that submitted the job (job originating user) or not (see section 8). However, the
1822 Printer object MUST respond with the 'unknown' value for any supported attribute (including all
1823 REQUIRED attributes) for which the Printer object does not know the value, unless it would
1824 violate the security policy. See the description of the "out-of-band" values in the beginning of
1825 Section 4.1.

1826

1827 Jobs are returned in the following order:

- 1828 - If the client requests all 'completed' Jobs (Jobs in the 'completed', 'aborted', or 'canceled'
1829 states), then the Jobs are returned newest to oldest (with respect to actual completion
1830 time)
- 1831 - If the client requests all 'not-completed' Jobs (Jobs in the 'pending', 'processing', 'pending-
1832 held', and 'processing-stopped' states), then Jobs are returned in relative chronological
1833 order of expected time to complete (based on whatever scheduling algorithm is
1834 configured for the Printer object).

1835

1836 3.2.7 Pause-Printer Operation

1837 This OPTIONAL operation allows a client to stop the Printer object from scheduling jobs on all its
1838 devices. Depending on implementation, the Pause-Printer operation MAY also stop the Printer from
1839 processing the current job or jobs. Any job that is currently being printed is either stopped as soon as the
1840 implementation permits or is completed, depending on implementation. The Printer object MUST still
1841 accept create operations to create new jobs, but MUST prevent any jobs from entering the 'processing'
1842 state.

1843 If the Pause-Printer operation is supported, then the Resume-Printer operation MUST be supported, and
1844 vice-versa.

1845 The IPP Printer stops the current job(s) on its device(s) that were in the 'processing' or 'processing-
1846 stopped' states as soon as the implementation permits. If the implementation ~~supports the "printer-state-
1847 reasons" attribute and the devices-~~ Issue 30 will take appreciable time to stop, the IPP Printer adds the
1848 'moving-to-paused' value to the Printer object's "printer-state-reasons" attribute (see section 4.4.12).
1849 When the device(s) have all stopped, the IPP Printer transitions the Printer object to the 'stopped' state,
1850 removes the 'moving-to-paused' value, if present, and adds the 'paused' value to the Printer object's
1851 "printer-state-reasons" attribute.

1852 When the current job(s) complete that were in the 'processing' state, the IPP Printer transitions them to
1853 the 'completed' state. When the current job(s) stop in mid processing that were in the 'processing' state,

1854 the IPP Printer transitions them to the 'processing-stopped' state and, ~~if the "job-state-reasons" attribute is~~
 1855 ~~supported,~~ **Issue 30** adds the 'printer-stopped' value to the job's "job-state-reasons" attribute.

1856 Note: for any jobs that are 'pending' or 'pending-held', the 'printer-stopped' value of the jobs' "job-state-
 1857 reasons" attribute also applies. However, the IPP Printer NEED NOT update those jobs' "job-state-
 1858 reasons" attributes and only need return the 'printer-stopped' value when those jobs are queried (so-called
 1859 "lazy evaluation").

1860 Whether the Pause-Printer operation affects jobs that were submitted to the device from other sources
 1861 than the IPP Printer object in the same way that the Pause-Printer operation affects jobs that were
 1862 submitted to the IPP Printer object using IPP, depends on implementation, i.e., on whether the IPP
 1863 protocol is being used as a universal management protocol or just to manage IPP jobs, respectively.

1864 The IPP Printer MUST accept the request in any state and transition the Printer to the indicated new
 1865 "printer-state" before returning as follows:

Current "printer-state"	New "printer-state"	"printer- state- reasons"	IPP Printer's response status code and action:
'idle'	'stopped'	'paused'	'successful-ok'
'processing'	'processing'	'moving-to- paused'	OPTION 1: 'successful-ok'; Later, when all output has stopped, the "printer- state" becomes 'stopped', and the 'paused' value replaces the 'moving-to-paused' value in the "printer-state-reasons" attribute
'processing'	'stopped'	'paused'	OPTION 2: 'successful-ok'; all device output stopped immediately
'stopped'	'stopped'	'paused'	'successful-ok'

1866 *Access Rights:* The authenticated user (see section 8.3) performing this operation ~~requesting user~~ must
 1867 be an operator or administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP Printer
 1868 MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or 'client-
 1869 error-not-authorized' as appropriate.

1870 3.2.7.1 Pause-Printer Request

1871 The following groups of attributes are part of the Pause-Printer Request:

1872 Group 1: Operation Attributes

1873 Natural Language and Character Set:

1874 The "attributes-charset" and "attributes-natural-language" attributes as described in section
 1875 3.1.4.1.

1876

1877 Target:

1878 The "printer-uri" (uri) operation attribute which is the target for this operation as described in
 1879 section 3.1.5.

1880
 1881 Requesting User Name:
 1882 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
 1883 described in section 8.3.

1884 3.2.7.2 Pause-Printer Response

1885 The following groups of attributes are part of the Pause-Printer Response:

1886 Group 1: Operation Attributes

1887 Status Message:

1888 In addition to the REQUIRED status code returned in every response, the response
 1889 OPTIONALLY includes a "status-message" (text) operation attribute as described in sections [13](#)
 1890 [and 3.1.6](#).

1891 Natural Language and Character Set:

1892 The "attributes-charset" and "attributes-natural-language" attributes as described in section
 1893 3.1.4.2.

1894 Group 2: Unsupported Attributes

1895
 1896 [See section 3.1.7 for details on returning Unsupported Attributes.](#)

1897 ~~This is a set of Operation attributes supplied by the client (in the request) that are not supported~~
 1898 ~~by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16).~~
 1899

1900 3.2.8 Resume-Printer Operation

1901 This operation allows a client to resume the Printer object scheduling jobs on all its devices. ~~If~~The
 1902 Printer object ~~supports the "printer-state-reasons" attribute, it~~ [Issue 30](#) MUST remove the 'paused' and
 1903 'moving-to-paused' values from the Printer object's "printer-state-reasons" attribute, if present. If there
 1904 are no other reasons to keep a device paused (such as media-jam), the IPP Printer transitions itself to the
 1905 'processing' or 'idle' states, depending on whether there are jobs to be processed or not, respectively, and
 1906 the device(s) resume processing jobs.

1907 If the Pause-Printer operation is supported, then the Resume-Printer operation MUST be supported, and
 1908 vice-versa.

1909 The IPP Printer removes the 'printer-stopped' value from any job's "job-state-reasons" attributes
 1910 contained in that Printer.

1911 The IPP Printer MUST accept the request in any state, transition the Printer object to the indicated new
 1912 state as follows:

Current "printer-state"	New "printer-state"	IPP Printer's response status code and action:
'idle'	'idle'	'successful-ok'
'processing'	'processing'	'successful-ok'

Current "printer-state"	New "printer-state"	IPP Printer's response status code and action:
'stopped'	'processing'	'successful-ok'; when there are jobs to be processed
'stopped'	'idle'	'successful-ok'; when there are no jobs to be processed.

1913 *Access Rights:* The [authenticated user \(see section 8.3\) performing this operation](#) ~~requesting user~~ must
 1914 be an operator or administrator of the Printer object ([see Sections 1 and 8.5](#)). Otherwise, the IPP Printer
 1915 MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or 'client-
 1916 error-not-authorized' as appropriate.

1917 The Resume-Printer Request and Resume-Printer Response have the same attribute groups and attributes
 1918 as the Pause-Printer operation (see sections 3.2.7.1 and 3.2.7.2).

1919 3.2.9 Purge-Jobs Operation

1920 This OPTIONAL operation allows a client to remove all jobs from an IPP Printer object, regardless of
 1921 their job states, including jobs in the Printer object's Job History (see Section 4.3.7.1). After a Purge-
 1922 Jobs operation has been performed, a Printer object MUST return no jobs in subsequent Get-Job-
 1923 Attributes and Get-Jobs responses (until new jobs are submitted).

1924 Whether the Purge-Jobs (and Get-Jobs) operation affects jobs that were submitted to the device from
 1925 other sources than the IPP Printer object in the same way that the Purge-Jobs operation affects jobs that
 1926 were submitted to the IPP Printer object using IPP, depends on implementation, i.e., on whether the IPP
 1927 protocol is being used as a universal management protocol or just to manage IPP jobs, respectively.

1928 Note: if an operator wants to cancel all jobs without clearing out the Job History, the operator uses the
 1929 Cancel-Job operation on each job instead of using the Purge-Job operation.

1930 The Printer object MUST accept this operation in any state and transition the Printer object to the 'idle'
 1931 state.

1932 *Access Rights:* The [authenticated user \(see section 8.3\) performing this operation](#) ~~requesting user~~ must
 1933 be an operator or administrator of the Printer object ([see Sections 1 and 8.5](#)). Otherwise, the IPP object
 1934 MUST reject the operation and return: client-error-forbidden, client-error-not-authenticated, and client-
 1935 error-not-authorized as appropriate.

1936 The Purge-Jobs Request and Purge-Jobs Response have the same attribute groups and attributes as the
 1937 Pause-Printer operation (see sections 3.2.7.1 and 3.2.7.2).

1938 3.3 Job Operations

1939 All Job operations are directed at Job objects. A client MUST always supply some means of identifying
 1940 the Job object in order to identify the correct target of the operation. That job identification MAY either

1941 be a single Job URI or a combination of a Printer URI with a Job ID. The IPP object implementation
1942 MUST support both forms of identification for every job.

1943 3.3.1 Send-Document Operation

1944 This OPTIONAL operation allows a client to create a multi-document Job object that is initially "empty"
1945 (contains no documents). In the Create-Job response, the Printer object returns the Job object's URI (the
1946 "job-uri" attribute) and the Job object's 32-bit identifier (the "job-id" attribute). For each new document
1947 that the client desires to add, the client uses a Send-Document operation. Each Send-Document Request
1948 contains the entire stream of document data for one document.

1949 If the Printer supports this operation but does not support multiple documents per job, the Printer MUST
1950 reject subsequent Send-Document operations supplied with data and return the 'server-error-multiple-
1951 document-jobs-not-supported'. However, the Printer MUST accept the first document with a 'true' or
1952 'false' value for the "last-document" operation attribute (see below), so that clients MAY always submit
1953 one document jobs with a 'false' value for "last-document" in the first Send-Document and a 'true' for
1954 "last-document" in the second Send-Document (with no data). Issue 34

1955 Since the Create-Job and the send operations (Send-Document or Send-URI operations) that follow
1956 could occur over an arbitrarily long period of time for a particular job, a client MUST send another send
1957 operation within an IPP Printer defined minimum time interval after the receipt of the previous request
1958 for the job. If a Printer object supports multiple document jobs, the Printer object MUST support the
1959 "multiple-operation-time-out" attribute (see section 4.4.31). This attribute indicates the minimum
1960 number of seconds the Printer object will wait for the next send operation before taking some recovery
1961 action.

1962 An IPP object MUST recover from an errant client that does not supply a send operation, sometime after
1963 the minimum time interval specified by the Printer object's "multiple-operation-time-out" attribute. Such
1964 recovery MAY include any of the following or other recovery actions:

- 1965 1. Assume that the Job is an invalid job, start the process of changing the job state to 'aborted', add
1966 the 'aborted-by-system' value to the job's "job-state-reasons" attribute (see section 4.3.8), ~~if~~
1967 ~~supported,~~ Issue 30 and clean up all resources associated with the Job. In this case, if another
1968 send operation is finally received, the Printer responds with an "client-error-not-possible" or
1969 "client-error-not-found" depending on whether or not the Job object is still around when the send
1970 operation finally arrives.
- 1971 2. Assume that the last send operation received was in fact the last document (as if the "last-
1972 document" flag had been set to 'true'), close the Job object, and proceed to process it (i.e., move
1973 the Job's state to 'pending').
- 1974 3. Assume that the last send operation received was in fact the last document, close the Job, but
1975 move it to the 'pending-held' and add the 'submission-interrupted' value to the job's "job-state-
1976 reasons" attribute (see section 4.3.8), ~~if supported.~~ Issue 30 This action allows the user or an
1977 operator to determine whether to continue processing the Job by moving it back to the 'pending'
1978 state using the Release-Job operation (see section 3.3.6) or to cancel the job using the Cancel-Job
1979 operation (see section 3.3.3).

1980

1981 Each implementation is free to decide the "best" action to take depending on local policy, whether any
1982 documents have been added, whether the implementation spools jobs or not, and/or any other piece of
1983 information available to it. If the choice is to abort the Job object, it is possible that the Job object may
1984 already have been processed to the point that some media sheet pages have been printed.

1985 Access Rights: The authenticated user (see section 8.3) performing this operation must either be the job
1986 owner (as determined in the Create-Job operation) or an operator or administrator of the Printer object
1987 (see Sections 1 and 8.5). Otherwise, the IPP object MUST reject the operation and return: 'client-error-
1988 forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' as appropriate. Issue 19

1989 3.3.1.1 Send-Document Request

1990 The following attribute sets are part of the Send-Document Request:

1991 Group 1: Operation Attributes

1992 Natural Language and Character Set:

1993 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1994 3.1.4.1.

1995 Target:

1996 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation
1997 attribute(s) which define the target for this operation as described in section 3.1.5.

1998 Requesting User Name:

1999 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
2000 described in section 8.3.

2001 "document-name" (name(MAX)):

2002 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
2003 attribute. It contains the client supplied document name. The document name MAY be different
2004 than the Job name. It might be helpful, but NEED NOT be unique across multiple documents in
2005 the same Job. Typically, the client software automatically supplies the document name on behalf
2006 of the end user by using a file name or an application generated name. See the description of the
2007 "document-name" operation attribute in the Print-Job Request (section 3.2.1.1) for more
2008 information about this attribute.

2009 "compression" (type3 keyword)

2010 ~~See the description of "compression" for the Print-Job operation in Section 3.2.1.1. The client~~
2011 ~~OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this attribute~~
2012 ~~and the "compression-supported" attribute (see section 4.4.29). The client supplied~~
2013 ~~"compression" operation attribute identifies the compression algorithm used on the document~~
2014 ~~data. If the client omits this attribute, the Printer object MUST assume that the data is not~~
2015 ~~compressed. If the client supplies the attribute and the Printer object supports the attribute, the~~
2016 ~~Printer object MUST use the corresponding decompression algorithm on the document data. If~~
2017 ~~the client supplies this attribute, but the value is not supported by the Printer object, i.e., the value~~

2022 ~~is not one of the values of the Printer object's "compression-supported" attribute, the Printer~~
2023 ~~object MUST copy the attribute and its value to the Unsupported Attributes response group,~~
2024 ~~reject the request, and return the 'client-error-attributes-or-values-not-supported' status code.~~

2025
2026

2027 "document-format" (mimeType) :

2028 ~~See the description of "document-format" for the Print-Job operation in Section 3.2.1.1. The~~
2029 ~~client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute.~~
2030 ~~The value of this attribute identifies the format of the supplied document data. If the client does~~
2031 ~~not supply this attribute, the Printer object assumes that the document data is in the format~~
2032 ~~defined by the Printer object's "document-format-default" attribute. If the client supplies this~~
2033 ~~attribute, but the value is not supported by the Printer object, i.e., the value is not one of the~~
2034 ~~values of the Printer object's "document-format-supported" attribute, the Printer object MUST~~
2035 ~~reject the request and return the 'client-error-document-format-not-supported' status code.~~ Issue
2036 11

2037

2038 "document-natural-language" (naturalLanguage):

2039 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports
2040 this attribute. This attribute specifies the natural language of the document for those document-
2041 formats that require a specification of the natural language in order to image the document
2042 unambiguously. There are no particular values required for the Printer object to support.

2043

2044

2045 "last-document" (boolean):

2046 The client MUST supply this attribute. The Printer object MUST support this attribute. It is a
2047 boolean flag that is set to 'true' if this is the last document for the Job, 'false' otherwise.

2048

2049 Group 2: Document Content

2050 The client MUST supply the document data if the "last-document" flag is set to 'false'. However,
2051 since a client might not know that the previous document sent with a Send-Document (or Send-
2052 URI) operation was the last document (i.e., the "last-document" attribute was set to 'false'), it is
2053 legal to send a Send-Document request with no document data where the "last-document" flag is
2054 set to 'true'. Such a request MUST NOT increment the value of the Job object's "number-of-
2055 documents" attribute, since no real document was added to the job.

2056 3.3.1.2 Send-Document Response

2057 The following sets of attributes are part of the Send-Document Response:

2058 Group 1: Operation Attributes

2059 Status Message:

2060 In addition to the REQUIRED status code returned in every response, the response
2061 OPTIONALLY includes a "status-message" (text) operation attribute as described in sections 13
2062 and 3.1.6.

2063
2064
2065
2066
2067

Natural Language and Character Set:

The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.2.

Group 2: Unsupported Attributes

~~See section 3.1.7 for details on returning Unsupported Attributes. This is a set of Operation attributes supplied by the client (in the request) that are not supported by the Printer object or that conflict with one another (see sections 3.2.1.2 and the Implementer's Guide [IPP-IG]). If the Printer object is not returning any Unsupported Attributes in the response, the Printer object SHOULD omit Group 2 rather than sending an empty group. However, a client MUST be able to accept an empty group.~~

Group 3: Job Object Attributes

This is the same set of attributes as described in the Print-Job response (see section 3.2.1.2).

3.3.2 Send-URI Operation

This OPTIONAL operation is identical to the Send-Document operation (see section 3.3.1) except that a client MUST supply a URI reference ("document-uri" operation attribute) rather than the document data itself. If a Printer object supports this operation, clients can use both Send-URI or Send-Document operations to add new documents to an existing multi-document Job object. However, if a client needs to indicate that the previous Send-URI or Send-Document was the last document, the client MUST use the Send-Document operation with no document data and the "last-document" flag set to 'true' (rather than using a Send-URI operation with no "document-uri" operation attribute).

If a Printer object supports this operation, it MUST also support the Print-URI operation (see section 3.2.2).

The Printer object MUST validate the syntax and URI scheme of the supplied URI before returning a response, just as in the Print-URI operation.

3.3.3 Cancel-Job Operation

This REQUIRED operation allows a client to cancel a Print Job from the time the job is created up to the time it is completed, canceled, or aborted. Since a Job might already be printing by the time a Cancel-Job is received, some media sheet pages might be printed before the job is actually terminated.

The IPP object MUST accept or reject the request based on the job's current state and transition the job to the indicated new state as follows:

<u>Current "job-state"</u>	<u>New "job-state"</u>	<u>IPP object's response status code and action:</u>
<u>'pending'</u>	<u>'canceled'</u>	<u>'successful-ok'</u>

<u>Current "job-state"</u>	<u>New "job-state"</u>	<u>IPP object's response status code and action:</u>
<u>'pending-held'</u>	<u>'canceled'</u>	<u>'successful-ok'</u>
<u>'processing'</u>	<u>'canceled'</u>	<u>'successful-ok'</u>
<u>'processing'</u>	<u>'processing'</u>	<u>'successful-ok' See Note 1</u>
<u>'processing'</u>	<u>'processing'</u>	<u>'client-error-not-possible' See Note 2</u>
<u>'processing-stopped'</u>	<u>'canceled'</u>	<u>'successful-ok'</u>
<u>'processing-stopped'</u>	<u>'processing-stopped'</u>	<u>'successful-ok' See Note 1</u>
<u>'processing-stopped'</u>	<u>'processing-stopped'</u>	<u>'client-error-not-possible' See Note 2</u>
<u>'completed'</u>	<u>'completed'</u>	<u>'client-error-not-possible'</u>
<u>'canceled'</u>	<u>'canceled'</u>	<u>'client-error-not-possible'</u>
<u>'aborted'</u>	<u>'aborted'</u>	<u>'client-error-not-possible'</u>

2097 Note 1: If the OPTIONAL "job-state-reasons" attribute is supported and if the implementation requires
 2098 some measurable time to cancel the job in the 'processing' or 'processing-stopped' job states, the IPP
 2099 object MUST add the 'processing-to-stop-point' value to the job's "job-state-reasons" attribute and then
 2100 transition the job to the 'canceled' state when the processing ceases (see section 4.3.8).

2101 Note 2: If the Job object already has the 'processing-to-stop-point' value in its "job-state-reasons"
 2102 attribute, then the Printer object MUST reject a Cancel-Job operation.

2103 Access Rights: The authenticated user (see section 8.3) performing this operation must either be the job
 2104 owner or an operator or administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP
 2105 object MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or
 2106 'client-error-not-authorized' as appropriate.

2107 3.3.3.1 Cancel-Job Request

2108 The following groups of attributes are part of the Cancel-Job Request:

2109 Group 1: Operation Attributes

2110 Natural Language and Character Set:

2111 The "attributes-charset" and "attributes-natural-language" attributes as described in section
 2112 3.1.4.1.

2113

2114 Target:

2115 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri)
 2116 operation attribute(s) which define the target for this operation as described in section 3.1.5.

2117

2118 Requesting User Name:

2119 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
 2120 described in section 8.3.

2121

2122 "message" (text(127)):

2123 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports
2124 this attribute. It is a message to the operator. This "message" attribute is not the same as the "job-
2125 message-from-operator" attribute. That attribute is used to report a message from the operator to
2126 the end user that queries that attribute. This "message" operation attribute is used to send a
2127 message from the client to the operator along with the operation request. It is an implementation
2128 decision of how or where to display this message to the operator (if at all).

2129

2130 3.3.3.2 Cancel-Job Response

2131 The following sets of attributes are part of the Cancel-Job Response:

2132 Group 1: Operation Attributes

2133 Status Message:

2134 In addition to the REQUIRED status code returned in every response, the response
2135 OPTIONALLY includes a "status-message" (text) operation attribute as described in sections 13
2136 and 3.1.6.

2137

2138 ~~If the job is already in the 'completed', 'aborted', or 'canceled' state, or the 'process-to-stop-point'~~
2139 ~~value is set in the Job's "job-state-reasons" attribute, the Printer object MUST reject the request~~
2140 ~~and return the 'client-error-not-possible' error status code.~~

2141

2142 Natural Language and Character Set:

2143 The "attributes-charset" and "attributes-natural-language" attributes as described in section
2144 3.1.4.2.

2145

2146 Group 2: Unsupported Attributes

2147 ~~See section 3.1.7 for details on returning Unsupported Attributes. This is a set of Operation~~
2148 ~~attributes supplied by the client (in the request) that are not supported by the Printer object or that~~
2149 ~~conflict with one another (see section 3.2.1.2 and the Implementer's Guide [IPP-IG]). If the~~
2150 ~~Printer object is not returning any Unsupported Attributes in the response, the Printer object~~
2151 ~~SHOULD omit Group 2 rather than sending an empty group. However, a client MUST be able~~
2152 ~~to accept an empty group.~~

2153

2154 Once a successful response has been sent, the implementation guarantees that the Job will eventually end
2155 up in the 'canceled' state. Between the time of the Cancel-Job operation is accepted and when the job
2156 enters the 'canceled' job-state (see section 4.3.7), the "job-state-reasons" attribute SHOULD contain the '
2157 processing-to-stop-point' value which indicates to later queries that although the Job might still be
2158 'processing', it will eventually end up in the 'canceled' state, not the 'completed' state.

2159 3.3.4 Get-Job-Attributes Operation

2160 This REQUIRED operation allows a client to request the values of attributes of a Job object and it is
2161 almost identical to the Get-Printer-Attributes operation (see section 3.2.5). The only differences are that
2162 the operation is directed at a Job object rather than a Printer object, there is no "document-format"
2163 operation attribute used when querying a Job object, and the returned attribute group is a set of Job
2164 object attributes rather than a set of Printer object attributes.

2165 For Jobs, the possible names of attribute groups are:

- 2166 - 'job-template': ~~all~~ the subset of the Job Template attributes that apply to a Job object (the first
2167 column of the table in Section 4.2) that the implementation supports for Job objects.
 - 2168 - 'job-description': ~~all~~ the subset of the Job Description attributes specified in Section 4.3 that the
2169 implementation supports for Job objects.
 - 2170 - 'all': the special group 'all' that includes all ~~supported~~ attributes that the implementation supports for
2171 Job objects. **Issue 23**
- 2172

2173 Since a client MAY request specific attributes or named groups, there is a potential that there is some
2174 overlap. For example, if a client requests, 'job-name' and 'job-description', the client is actually
2175 requesting the "job-name" attribute once by naming it explicitly, and once by inclusion in the 'job-
2176 description' group. In such cases, the Printer object NEED NOT return the attribute only once in the
2177 response even if it is requested multiple times. The client SHOULD NOT request the same attribute in
2178 multiple ways.

2179 It is NOT REQUIRED that a Job object support all attributes belonging to a group (since some attributes
2180 are OPTIONAL). However it is REQUIRED that each Job object support all group names.

2181 3.3.4.1 Get-Job-Attributes Request

2182 The following groups of attributes are part of the Get-Job-Attributes Request when the request is
2183 directed at a Job object:

2184 Group 1: Operation Attributes

2185 Natural Language and Character Set:

2186 The "attributes-charset" and "attributes-natural-language" attributes as described in section
2187 3.1.4.1.

2188
2189 Target:

2190 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri)
2191 operation attribute(s) which define the target for this operation as described in section 3.1.5.

2192
2193 Requesting User Name:

2194 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
2195 described in section 8.3.

2196

2197 "requested-attributes" (1setOf keyword) :

2198 The client OPTIONALLY supplies this attribute. The IPP object MUST support this attribute.
2199 It is a set of attribute names and/or attribute group names in whose values the requester is
2200 interested. If the client omits this attribute, the IPP object MUST respond as if this attribute had
2201 been supplied with a value of 'all'.
2202

2203 3.3.4.2 Get-Job-Attributes Response

2204 The Printer object returns the following sets of attributes as part of the Get-Job-Attributes Response:

2205 Group 1: Operation Attributes

2206 Status Message:

2207 In addition to the REQUIRED status code returned in every response, the response
2208 OPTIONALLY includes a "status-message" (text) operation attribute as described in sections 13
2209 and 3.1.6.
2210

2211 Natural Language and Character Set:

2212 The "attributes-charset" and "attributes-natural-language" attributes as described in section
2213 3.1.4.2. The "attributes-natural-language" MAY be the natural language of the Job object, rather
2214 than the one requested.
2215

2216 Group 2: Unsupported Attributes

2217 See section 3.1.7 for details on returning Unsupported Attributes.

2218 ~~This is a set of Operation attributes supplied by the client (in the request) that are not supported
2219 by the Printer object or that conflict with one another (see sections 3.2.1.2 and the Implementer's
2220 Guide [IPP-IG]). The response NEED NOT contain the "requested-attributes" operation
2221 attribute with any supplied values (attribute keywords) that were requested by the client but are
2222 not supported by the IPP object. If the Printer does include unsupported attributes referenced in
2223 "requested-attributes" and such attributes include group names, such as 'all', the unsupported
2224 attributes MUST NOT include attributes described in the standard but not supported by the
2225 implementation. Issue 23 ~~If the Printer object is not returning any Unsupported Attributes in the
2226 response, the Printer object SHOULD omit Group 2 rather than sending an empty group.
2227 However, a client MUST be able to accept an empty group.~~
2228
2229~~

2230 Group 3: Job Object Attributes

2231 This is the set of requested attributes and their current values. The IPP object ignores (does not
2232 respond with) any requested attribute or value which is not supported or which is restricted by the
2233 security policy in force, including whether the requesting user is the user that submitted the job
2234 (job originating user) or not (see section 8). However, the IPP object MUST respond with the
2235 'unknown' value for any supported attribute (including all REQUIRED attributes) for which the

2236 IPP object does not know the value, unless it would violate the security policy. See the
2237 description of the "out-of-band" values in the beginning of Section 4.1.

2238 3.3.5 Hold-Job Operation

2239 This OPTIONAL operation allows a client to hold a pending job in the queue so that it is not eligible for
2240 scheduling. If the Hold-Job operation is supported, then the Release-Job operation MUST be supported,
2241 and vice-versa. The OPTIONAL "job-hold-until" operation attribute allows a client to specify whether
2242 to hold the job indefinitely or until a specified time period, if supported.

2243 The IPP object MUST accept or reject the request based on the job's current state and transition the job
2244 to the indicated new state as follows:

Current "job-state"	New "job-state"	IPP object's response status code and action:
'pending'	'pending-held'	'successful-ok' See Note 1
'pending'	'pending'	'successful-ok' See Note 2
'pending-held'	'pending-held'	'successful-ok' See Note 1
'pending-held'	'pending'	'successful-ok' See Note 2
'processing'	'processing'	'client-error-not-possible'
'processing-stopped'	'processing-stopped'	'client-error-not-possible'
'completed'	'completed'	'client-error-not-possible'
'canceled'	'canceled'	'client-error-not-possible'
'aborted'	'aborted'	'client-error-not-possible'

2245 Note 1: If ~~the OPTIONAL "job-state-reasons" attribute is supported and if~~ the implementation supports
2246 multiple reasons for a job to be in the 'pending-held' state, the IPP object MUST add the 'job-hold-until-
2247 specified' value to the job's "job-state-reasons" attribute.

2248 Note 2: If the IPP object supports the "job-hold-until" operation attribute, but the specified time period
2249 has already started (or is the 'no-hold' value) and there are no other reasons to hold the job, the IPP object
2250 MUST make the job be a candidate for processing immediately (see Section 4.2.2) by putting the job in
2251 the 'pending' state.

2252 Note: In order to keep the Hold-Job operation simple, such a request is rejected when the job is in the
2253 'processing' or 'processing-stopped' states. If an operation is needed to hold jobs while in these states, it
2254 will be added as an additional operation, rather than overloading the Hold-Job operation. Then it is clear
2255 to clients by querying the Printer object's "operations-supported" (see Section 4.4.15) and the Job
2256 object's "job-state" (see Section 4.3.7) attributes which operations are possible.

2257 *Access Rights:* The ~~requesting-authenticated~~ user (~~see section 8.3~~) ~~performing this operation~~ must either
2258 be the ~~submitter-of-the~~ job ~~owner~~ or an operator or administrator of the Printer object (see Sections 1 and
2259 8.5). Otherwise, the IPP object MUST reject the operation and return: 'client-error-forbidden', 'client-
2260 error-not-authenticated', or 'client-error-not-authorized' as appropriate.

2261 3.3.5.1 Hold-Job Request

2262 The groups and operation attributes are the same as for a Cancel-Job request (see section 3.3.3.1), with
2263 the addition of the following Group 1 Operation attribute:

2264 "job-hold-until" (type3 keyword | name(MAX)):

2265 The client **OPTIONALLY** supplies this Operation attribute. The IPP object **MUST** support this
2266 operation attribute in a Hold-Job request, if it supports the "job-hold-until" Job template attribute
2267 in create operations. See section 4.2.2. The IPP object **SHOULD** support the "job-hold-until"
2268 Job Template attribute for use in job create operations with at least the 'indefinite' value, if it
2269 supports the Hold-Job operation. Otherwise, a client cannot create a job and hold it immediately
2270 (without picking some supported time period in the future).

2271 If supplied and supported as specified in the Printer's "job-hold-until-supported" attribute, the IPP
2272 object copies the supplied operation attribute to the Job object, replacing the job's previous "job-
2273 hold-until" attribute, if present, and makes the job a candidate for scheduling during the supplied
2274 named time period.

2275 If supplied, but either the "job-hold-until" Operation attribute itself or the value supplied is not
2276 supported, the IPP object accepts the request, returns the unsupported attribute or value in the
2277 Unsupported Attributes Group according to section 3.1.73.2.1.2, returns the 'successful-ok-
2278 ignored-or-substituted-attributes, and holds the job indefinitely until a client performs a
2279 subsequent Release-Job operation.

2280 If the client (1) supplies a value that specifies a time period that has already started or the 'no-
2281 hold' value (meaning don't hold the job) and (2) the IPP object supports the "job-hold-until"
2282 operation attribute and there are no other reasons to hold the job, the IPP object **MUST** accept the
2283 operation and make the job be a candidate for processing immediately (see Section 4.2.2).

2284 If the client does not supply a "job-hold-until" Operation attribute in the request, the IPP object
2285 **MUST** populate the job object with a "job-hold-until" attribute with the 'indefinite' value (if IPP
2286 object supports the "job-hold-until" attribute) and hold the job indefinitely, until a client performs
2287 a Release-Job operation.

2288 3.3.5.2 Hold-Job Response

2289 The groups and attributes are the same as for a Cancel-Job response (see section 3.3.3.2).

2290 3.3.6 Release-Job Operation

2291 This **OPTIONAL** operation allows a client to release a previously held job so that it is again eligible for
2292 scheduling. If the Hold-Job operation is supported, then the Release-Job operation **MUST** be supported,
2293 and vice-versa.

2294 This operation removes the "job-hold-until" job attribute, if present, from the job object that had been
2295 supplied in the create or most recent Hold-Job or Restart-Job operation and remove its effect on the job.

2296 ~~If the OPTIONAL "job-state-reasons" attribute is supported,~~ **Issue 30** The IPP object MUST remove the
 2297 'job-hold-until-specified' value from the job's "job-state-reasons" attribute, if present. See section 4.3.8.

2298 The IPP object MUST accept or reject the request based on the job's current state and transition the job
 2299 to the indicated new state as follows:

Current "job-state"	New "job-state"	IPP object's response status code and action:
'pending'	'pending'	'successful-ok' No effect on the job.
'pending-held'	'pending-held'	'successful-ok' See Note 1
'pending-held'	'pending'	'successful-ok'
'processing'	'processing'	'successful-ok' No effect on the job.
'processing-stopped'	'processing-stopped'	'successful-ok' No effect on the job.
'completed'	'completed'	'client-error-not-possible'
'canceled'	'canceled'	'client-error-not-possible'
'aborted'	'aborted'	'client-error-not-possible'

2300 Note 1: If there are other reasons to keep the job in the 'pending-held' state, such as 'resources-are-not-
 2301 ready', the job remains in the 'pending-held' state. Thus the 'pending-held' state is not just for jobs that
 2302 have the 'job-hold-until' applied to them, but are for any reason to keep the job from being a candidate
 2303 for scheduling and processing, such as 'resources-are-not-ready'. See the "job-hold-until" attribute
 2304 (section 4.2.2).

2305 *Access Rights:* The [authenticated user \(see section 8.3\) performing this operation requesting user](#) must
 2306 either be the [submitter of the job owner](#) or an operator or administrator of the Printer object ([see Sections](#)
 2307 [1 and 8.5](#)). Otherwise, the IPP object MUST reject the operation and return: 'client-error-forbidden',
 2308 'client-error-not-authenticated', or 'client-error-not-authorized' as appropriate.

2309 The Release-Job Request and Release-Job Response have the same attribute groups and attributes as the
 2310 Cancel-Job operation (see section 3.3.3.1 and 3.3.3.2).

2311 3.3.7 Restart-Job Operation

2312 This OPTIONAL operation allows a client to restart a job that is retained in the queue after processing
 2313 has completed (see section 4.3.7.1).

2314 The job is moved to the 'pending' job state and restarts at the beginning on the same IPP Printer object
 2315 with the same attribute values. The Job Description attributes that accumulate job progress, such as
 2316 "job-impressions-completed", "job-media-sheets-completed", and "job-k-octets-processed", MUST be
 2317 reset to 0 so that they give an accurate record of the job from its restart point. The job object MUST
 2318 continue to use the same "job-uri" and "job-id" attribute values.

2319 Note: If in the future an operation is needed that does not reset the job progress attributes, then a new
 2320 operation will be defined which makes a copy of the job, assigns a new "job-uri" and "job-id" to the copy
 2321 and resets the job progress attributes in the new copy only.

2322 The IPP object MUST accept or reject the request based on the job's current state, transition the job to
 2323 the indicated new state as follows:

Current "job-state"	New "job-state"	IPP object's response status code and action:
'pending'	'pending'	'client-error-not-possible'.
'pending-held'	'pending-held'	'client-error-not-possible'.
'processing'	'processing'	'client-error-not-possible'.
'processing-stopped'	<u>'processing-stopped'</u>	<u>'client-error-not-possible'</u> .
'completed'	'pending'	'successful-ok' - job is started over.
'completed'	'completed'	'client-error-not-possible' - see Note 1
'canceled'	'pending'	'successful-ok' - job is started over.
'canceled'	'canceled'	'client-error-not-possible' - see Note 1
'aborted'	'pending'	'successful-ok' - job is started over.
'aborted'	'aborted'	'client-error-not-possible' - see Note 1

2324

2325 Note 1: If the Job Retention Period has expired for the job in this state, then the IPP object rejects the
 2326 operation. See section 4.3.7.1.

2327 Note: In order to prevent a user from inadvertently restarting a job in the middle, the Restart-Job request
 2328 is rejected when the job is in the 'processing' or 'processing-stopped' states. If in the future an operation
 2329 is needed to hold or restart jobs while in these states, it will be added as an additional operation, rather
 2330 than overloading the Restart-Job operation, so that it is clear that the user intended that the current job
 2331 not be completed.

2332 *Access Rights:* The authenticated user (see section 8.3) performing this operation ~~requesting user~~ must
 2333 either be the ~~submitter of the~~ job owner or an operator or administrator of the Printer object (see Sections
 2334 1 and 8.5). Otherwise, the IPP object MUST reject the operation and return: 'client-error-forbidden',
 2335 'client-error-not-authenticated', or 'client-error-not-authorized' as appropriate.

2336 3.3.7.1 Restart-Job Request

2337 The groups and attributes are the same as for a Cancel-Job request (see section 3.3.3.1), with the addition
 2338 of the following Group 1 Operation attribute:

2339 "job-hold-until" (type3 keyword | name(MAX)):

2340 The client OPTIONALLY supplies this attribute. The IPP object MUST support this Operation
 2341 attribute in a Restart-Job request, if it supports the "job-hold-until" Job Template attribute in
 2342 create operations. See section 4.2.2. Otherwise, the IPP object NEED NOT support the "job-
 2343 hold-until" Operation attribute in a Restart-Job request.

2344 If supplied and supported as specified in the Printer's "job-hold-until-supported" attribute, the IPP
 2345 object copies the supplied Operation attribute to the Job object, replacing the job's previous "job-
 2346 hold-until" attribute, if present, and makes the job a candidate for scheduling during the supplied
 2347 named time period. See section 4.2.2.

2348 If supplied, but the value is not supported, the IPP object accepts the request, returns the
2349 unsupported attribute or value in the Unsupported Attributes Group according to section
2350 3.1.73.2.1.2, returns the 'successful-ok-ignored-or-substituted-attributes' status code, and holds
2351 the job indefinitely until a client performs a subsequent Release-Job operation.

2352 If supplied, but the "job-hold-until" Operation attribute itself is not supported, the IPP object
2353 accepts the request, returns the unsupported attribute with the out-of-band 'unsupported' value in
2354 the Unsupported Attributes Group according to section 3.1.73.2.1.2, returns the 'successful-ok-
2355 ignored-or-substituted-attributes' status code, and restarts the job, i.e., ignores the "job-hold-
2356 until" attribute.

2357 If the client (1) supplies a value that specifies a time period that has already started or the 'no-
2358 hold' value (meaning don't hold the job) and (2) the IPP object supports the "job-hold-until"
2359 operation attribute and there are no other reasons to hold the job, the IPP object makes the job a
2360 candidate for processing immediately (see Section 4.2.2).

2361 If the client does not supply a "job-hold-until" operation attribute in the request, the IPP object
2362 removes the "job-hold-until" attribute, if present, from the job. If there are no other reasons to
2363 hold the job, the Restart-Job operation makes the job a candidate for processing immediately (see
2364 Section 4.2.2).

2365 3.3.7.2 Restart-Job Response

2366 The groups and attributes are the same as for a Cancel-Job response (see section 3.3.3.2).

2367 Note: In the future an OPTIONAL Modify-Job operation may be specified that allows the client to
2368 modify other attributes before releasing the restarted job.

2369 4. Object Attributes

2370 This section describes the attributes with their corresponding attribute syntaxes and values that are part
2371 of the IPP model. The sections below show the objects and their associated attributes which are
2372 included within the scope of this protocol. Many of these attributes are derived from other relevant
2373 specifications:

- 2374 - Document Printing Application (DPA) [ISO10175]
- 2375 - RFC 1759 Printer MIB [RFC1759]

2376

2377 Each attribute is uniquely identified in this document using a "keyword" (see section 12.2.1) which is the
2378 name of the attribute. The keyword is included in the section header describing that attribute.

2379 Note: Not only are keywords used to identify attributes, but one of the attribute syntaxes described
2380 below is "keyword" so that some attributes have keyword values. Therefore, these attributes are defined
2381 as having an attribute syntax that is a set of keywords.

2382 4.1 Attribute Syntaxes

2383 This section defines the basic attribute syntax types that all clients and IPP objects MUST be able to
2384 accept in responses and accept in requests, respectively. Each attribute description in sections 3 and
2385 3.3.5 includes the name of attribute syntax(es) in the heading (in parentheses). A conforming
2386 implementation of an attribute MUST include the semantics of the attribute syntax(es) so identified.
2387 Section 6.3 describes how the protocol can be extended with new attribute syntaxes.

2388 The attribute syntaxes are specified in the following sub-sections, where the sub-section heading is the
2389 keyword name of the attribute syntax inside the single quotes. In operation requests and responses each
2390 attribute value MUST be represented as one of the attribute syntaxes specified in the sub-section heading
2391 for the attribute. In addition, the value of an attribute in a response (but not in a request) MAY be one of
2392 the "out-of-band" values [whose special encoding rules are defined in the "Encoding and Transport"](#)
2393 [specification \[IPP-PRO\]](#). Standard "out-of-band" values are: [Issue 12 and Issue 15](#)

2394 `unknown`: The attribute is supported by the IPP object, but the value is unknown to the IPP object
2395 for some reason.

2396 `unsupported`: The attribute is unsupported by the IPP object. This value MUST be returned only as
2397 the value of an attribute in the Unsupported Attributes Group.

2398 `no-value`: The attribute is supported by the Printer object, but the administrator has not yet
2399 configured a value.

2400

2401 ~~The "Encoding and Transport" specification [IPP-PRO] defines mechanisms for passing "out-of-band"~~
2402 ~~values.~~ All attributes in a request MUST have one or more values as defined in Sections 4.2 to 4.4.
2403 Thus clients MUST NOT supply attributes with "out-of-band" values. All attributes in a response
2404 MUST have one or more values as defined in Sections 4.2 to 4.4 or a single "out-of-band" value.

2405 Most attributes are defined to have a single attribute syntax. However, a few attributes (e.g., "job-sheet",
2406 "media", "job-hold-until") are defined to have several attribute syntaxes, depending on the value. These
2407 multiple attribute syntaxes are separated by the "|" character in the sub-section heading to indicate the
2408 choice. Since each value MUST be tagged as to its attribute syntax in the protocol, a single-valued
2409 attribute instance may have any one of its attribute syntaxes and a multi-valued attribute instance may
2410 have a mixture of its defined attribute syntaxes.

2411 4.1.1 `text`

2412 A text attribute is an attribute whose value is a sequence of zero or more characters encoded in a
2413 maximum of 1023 (MAX) octets. MAX is the maximum length for each value of any text attribute.
2414 However, if an attribute will always contain values whose maximum length is much less than MAX, the
2415 definition of that attribute will include a qualifier that defines the maximum length for values of that
2416 attribute. For example: the "printer-location" attribute is specified as "printer-location (text(127))". In
2417 this case, text values for "printer-location" MUST NOT exceed 127 octets; if supplied with a longer text
2418 string via some external interface (other than the protocol), implementations are free to truncate to this
2419 shorter length limitation.

2420 In this specification, all text attributes are defined using the 'text' syntax. However, 'text' is used only for
2421 brevity; the formal interpretation of 'text' is: 'textWithoutLanguage | textWithLanguage'. That is, for any
2422 attribute defined in this specification using the 'text' attribute syntax, all IPP objects and clients MUST
2423 support both the 'textWithoutLanguage' and 'textWithLanguage' attribute syntaxes. However, in actual
2424 usage and protocol execution, objects and clients accept and return only one of the two syntax per
2425 attribute. The syntax 'text' never appears "on-the-wire".

2426 Both 'textWithoutLanguage' and 'textWithLanguage' are needed to support the real world needs of
2427 interoperability between sites and systems that use different natural languages as the basis for human
2428 communication. Generally, one natural language applies to all text attributes in a given request or
2429 response. The language is indicated by the "attributes-natural-language" operation attribute defined in
2430 section 3.1.4 or "attributes-natural-language" job attribute defined in section 4.3.18, and there is no need
2431 to identify the natural language for each text string on a value-by-value basis. In these cases, the
2432 attribute syntax 'textWithoutLanguage' is used for text attributes. In other cases, the client needs to
2433 supply or the Printer object needs to return a text value in a natural language that is different from the
2434 rest of the text values in the request or response. In these cases, the client or Printer object uses the
2435 attribute syntax 'textWithLanguage' for text attributes (this is the Natural Language Override mechanism
2436 described in section 3.1.4).

2437 The 'textWithoutLanguage' and 'textWithLanguage' attribute syntaxes are described in more detail in the
2438 following sections.

2439 4.1.1.1 'textWithoutLanguage'

2440 The 'textWithoutLanguage' syntax indicates a value that is sequence of zero or more characters. Text
2441 strings are encoded using the rules of some charset. The Printer object MUST support the UTF-8
2442 charset [RFC2279] and MAY support additional charsets to represent 'text' values, provided that the
2443 charsets are registered with IANA [IANA-CS]. See Section 4.1.7 for the specification of the 'charset'
2444 attribute syntax, including restricted semantics and examples of charsets.

2445 4.1.1.2 'textWithLanguage'

2446 The 'textWithLanguage' attribute syntax is a compound attribute syntax consisting of two parts: a
2447 'textWithoutLanguage' part plus an additional 'naturalLanguage' (see section 4.1.8) part that overrides the
2448 natural language in force. The 'naturalLanguage' part explicitly identifies the natural language that
2449 applies to the text part of that value and that value alone. For any give text attribute, the
2450 'textWithoutLanguage' part is limited to the maximum length defined for that attribute, but the
2451 'naturalLanguage' part is always limited to 63 octets. Using the 'textWithLanguage' attribute syntax rather
2452 than the normal 'textWithoutLanguage' syntax is the so-called Natural Language Override mechanism
2453 and MUST be supported by all IPP objects and clients.

2454 If the attribute is multi-valued (1setOf text), then the 'textWithLanguage' attribute syntax MUST be used
2455 to explicitly specify each attribute value whose natural language needs to be overridden. Other values in
2456 a multi-valued 'text' attribute in a request or a response revert to the natural language of the operation
2457 attribute.

2458 In a create request, the Printer object MUST accept and store with the Job object any natural language in
2459 the "attributes-natural-language" operation attribute, whether the Printer object supports that natural
2460 language or not. Furthermore, the Printer object MUST accept and store any 'textWithLanguage'
2461 attribute value, whether the Printer object supports that natural language or not. These requirements are
2462 independent of the value of the "ipp-attribute-fidelity" operation attribute that the client MAY supply.

2463 Example: If the client supplies the "attributes-natural-language" operation attribute with the value: 'en'
2464 indicating English, but the value of the "job-name" attribute is in French, the client MUST use the
2465 'textWithLanguage' attribute syntax with the following two values:

2466 'fr': Natural Language Override indicating French

2467 'Rapport Mensuel': the job name in French

2468

2469 See the "Encoding and Transport" document [IPP-PRO] for a detailed example of the
2470 'textWithLanguage' attribute syntax.

2471 4.1.2 'name'

2472 This syntax type is used for user-friendly strings, such as a Printer name, that, for humans, are more
2473 meaningful than identifiers. Names are never translated from one natural language to another. The
2474 'name' attribute syntax is essentially the same as 'text', including the REQUIRED support of UTF-8
2475 except that the sequence of characters is limited so that its encoded form MUST NOT exceed 255
2476 (MAX) octets.

2477 Also like 'text', 'name' is really an abbreviated notation for either 'nameWithoutLanguage' or
2478 'nameWithLanguage'. That is, all IPP objects and clients MUST support both the
2479 'nameWithoutLanguage' and 'nameWithLanguage' attribute syntaxes. However, in actual usage and
2480 protocol execution, objects and clients accept and return only one of the two syntax per attribute. The
2481 syntax 'name' never appears "on-the-wire".

2482 Note: Only the 'text' and 'name' attribute syntaxes permit the Natural Language Override mechanism.

2483 Some attributes are defined as 'type3 keyword | name'. These attributes support values that are either
2484 type3 keywords or names. This dual-syntax mechanism enables a site administrator to extend these
2485 attributes to legally include values that are locally defined by the site administrator. Such names are not
2486 registered with IANA.

2487 4.1.2.1 'nameWithoutLanguage'

2488 The 'nameWithoutLanguage' syntax indicates a value that is sequence of zero or more characters so that
2489 its encoded form does not exceed MAX octets.

2490 4.1.2.2 'nameWithLanguage'

2491 The 'nameWithLanguage' attribute syntax is a compound attribute syntax consisting of two parts: a
2492 'nameWithoutLanguage' part plus an additional 'naturalLanguage' (see section 4.1.8) part that overrides

2493 the natural language in force. The 'naturalLanguage' part explicitly identifies the natural language that
2494 applies to that name value and that name value alone.

2495 The 'nameWithLanguage' attribute syntax behaves the same as the 'textWithLanguage' syntax. If a name
2496 is in a language that is different than the rest of the object or operation, then this 'nameWithLanguage'
2497 syntax is used rather than the generic 'nameWithoutLanguage' syntax.

2498 Example: If the client supplies the "attributes-natural-language" operation attribute with the value: 'en'
2499 indicating English, but the "printer-name" attribute is in German, the client MUST use the
2500 'nameWithLanguage' attribute syntax as follows:

2501 'de': Natural Language Override indicating German

2502 'Farbdrucker': the Printer name in German

2503

2504 4.1.2.3 Matching 'name' attribute values

2505 For purposes of matching two 'name' attribute values for equality, such as in job validation (where a
2506 client-supplied value for attribute "xxx" is checked to see if the value is among the values of the Printer
2507 object's corresponding "xxx-supported" attribute), the following match rules apply:

2508 1. 'keyword' values never match 'name' values.

2509 2. 'name' (nameWithoutLanguage and nameWithLanguage) values match if (1) the name parts
2510 match and (2) the Associated Natural-Language parts (see section 3.1.4.1) match. The matching
2511 rules are:

2512 a. the name parts match if the two names are identical character by character, except it is
2513 RECOMMENDED that case be ignored. For example: 'Ajax-letter-head-white' MUST
2514 match 'Ajax-letter-head-white' and SHOULD match 'ajax-letter-head-white' and 'AJAX-
2515 LETTER-HEAD-WHITE'.

2516 b. the Associated Natural-Language parts match if the shorter of the two meets the
2517 syntactic requirements of RFC 1766 [RFC1766] and matches byte for byte with the
2518 longer. For example, 'en' matches 'en', 'en-us' and 'en-gb', but matches neither 'fr' nor 'e'.

2519 4.1.3 'keyword'

2520 The 'keyword' attribute syntax is a sequence of characters, length: 1 to 255, containing only the US-
2521 ASCII [ASCII] encoded values for lowercase letters ("a" - "z"), digits ("0" - "9"), hyphen ("-"), dot ("."),
2522 and underscore ("_"). The first character MUST be a lowercase letter. Furthermore, keywords MUST
2523 be in U.S. English.

2524 This syntax type is used for enumerating semantic identifiers of entities in the abstract protocol, i.e.,
2525 entities identified in this document. Keywords are used as attribute names or values of attributes.
2526 Unlike 'text' and 'name' attribute values, 'keyword' values MUST NOT use the Natural Language
2527 Override mechanism, since they MUST always be US-ASCII and U.S. English.

2528 Keywords are for use in the protocol. A user interface will likely provide a mapping between protocol
2529 keywords and displayable user-friendly words and phrases which are localized to the natural language of
2530 the user. While the keywords specified in this document MAY be displayed to users whose natural
2531 language is U.S. English, they MAY be mapped to other U.S. English words for U.S. English users,
2532 since the user interface is outside the scope of this document.

2533 In the definition for each attribute of this syntax type, the full set of defined keyword values for that
2534 attribute are listed.

2535 When a keyword is used to represent an attribute (its name), it MUST be unique within the full scope of
2536 all IPP objects and attributes. When a keyword is used to represent a value of an attribute, it MUST be
2537 unique just within the scope of that attribute. That is, the same keyword MUST NOT be used for two
2538 different values within the same attribute to mean two different semantic ideas. However, the same
2539 keyword MAY be used across two or more attributes, representing different semantic ideas for each
2540 attribute. Section 6.1 describes how the protocol can be extended with new keyword values. Examples
2541 of attribute name keywords:

2542 "job-name"
2543 "attributes-charset"
2544

2545 Note: This document uses "type1", "type2", and "type3" prefixes to the "keyword" basic syntax to
2546 indicate different levels of review for extensions (see section 6.1).

2547 4.1.4 'enum'

2548 The 'enum' attribute syntax is an enumerated integer value that is in the range from 1 to 2**31 - 1
2549 (MAX). Each value has an associated 'keyword' name. In the definition for each attribute of this syntax
2550 type, the full set of possible values for that attribute are listed. This syntax type is used for attributes for
2551 which there are enum values assigned by other standards, such as SNMP MIBs. A number of attribute
2552 enum values in this specification are also used for corresponding attributes in other standards
2553 [RFC1759]. This syntax type is not used for attributes to which the administrator may assign values.
2554 Section 6.1 describes how the protocol can be extended with new enum values.

2555 Enum values are for use in the protocol. A user interface will provide a mapping between protocol enum
2556 values and displayable user-friendly words and phrases which are localized to the natural language of the
2557 user. While the enum symbols specified in this document MAY be displayed to users whose natural
2558 language is U.S. English, they MAY be mapped to other U.S. English words for U.S. English users,
2559 since the user interface is outside the scope of this document.

2560 Note: SNMP MIBs use '2' for 'unknown' which corresponds to the IPP "out-of-band" value 'unknown'.
2561 See the description of the "out-of-band" values at the beginning of Section 4.1. Therefore, attributes of
2562 type 'enum' start at '3'.

2563 Note: This document uses "type1", "type2", and "type3" prefixes to the "enum" basic syntax to indicate
2564 different levels of review for extensions (see section 6.1).

2565 4.1.5 'uri'

2566 The 'uri' attribute syntax is any valid Uniform Resource Identifier or URI [RFC2396]. Most often, URIs
2567 are simply Uniform Resource Locators or URLs. The maximum length of URIs used as values of IPP
2568 attributes is 1023 octets. Although most other IPP attribute syntax types allow for only lower-cased
2569 values, this attribute syntax type conforms to the case-sensitive and case-insensitive rules specified in
2570 [RFC2396]. See also [IPP-IIG] for a discussion of case in URIs.

2571 4.1.6 'uriScheme'

2572 The 'uriScheme' attribute syntax is a sequence of characters representing a URI scheme according to
2573 RFC 2396 [RFC2396]. Though RFC 2396 requires that the values be case-insensitive, IPP requires all
2574 lower case values in IPP attributes to simplify comparing by IPP clients and Printer objects. Standard
2575 values for this syntax type are the following keywords:

2576 'http': for HTTP schemed URIs (e.g., "http:....")
2577 'https': for use with HTTPS schemed URIs (e.g., "https:...") (not on IETF standards track)
2578 'ftp': for FTP schemed URIs (e.g., "ftp:...")
2579 'mailto': for SMTP schemed URIs (e.g., "mailto:...")
2580 'file': for file schemed URIs (e.g., "file:...")
2581

2582 A Printer object MAY support any URI 'scheme' that has been registered with IANA [IANA-MT]. The
2583 maximum length of URI 'scheme' values used to represent IPP attribute values is 63 octets.

2584 4.1.7 'charset'

2585 The 'charset' attribute syntax is a standard identifier for a charset. A charset is a coded character set and
2586 encoding scheme. Charsets are used for labeling certain document contents and 'text' and 'name'
2587 attribute values. The syntax and semantics of this attribute syntax are specified in RFC 2046 [RFC2046]
2588 and contained in the IANA character-set Registry [IANA-CS] according to the IANA procedures
2589 [RFC2278]. Though RFC 2046 requires that the values be case-insensitive US-ASCII, IPP requires all
2590 lower case values in IPP attributes to simplify comparing by IPP clients and Printer objects. When a
2591 character-set in the IANA registry has more than one name (alias), the name labeled as "(preferred
2592 MIME name)", if present, MUST be used.

2593 The maximum length of 'charset' values used to represent IPP attribute values is 63 octets.

2594 Some examples are:

2595 'utf-8': ISO 10646 Universal Multiple-Octet Coded Character Set (UCS) represented as the UTF-8
2596 [RFC2279] transfer encoding scheme in which US-ASCII is a subset charset.
2597 'us-ascii': 7-bit American Standard Code for Information Interchange (ASCII), ANSI X3.4-1986
2598 [ASCII]. That standard defines US-ASCII, but RFC 2045 [RFC2045] eliminates most of the
2599 control characters from conformant usage in MIME and IPP.

2600 `iso-8859-1': 8-bit One-Byte Coded Character Set, Latin Alphabet Nr 1 [ISO8859-1]. That standard
2601 defines a coded character set that is used by Latin languages in the Western Hemisphere and
2602 Western Europe. US-ASCII is a subset charset.

2603 `iso-10646-ucs-2': ISO 10646 Universal Multiple-Octet Coded Character Set (UCS) represented as
2604 two octets (UCS-2), with the high order octet of each pair coming first (so-called Big Endian
2605 integer).
2606

2607 Some attribute descriptions MAY place additional requirements on charset values that may be used, such
2608 as REQUIRED values that MUST be supported or additional restrictions, such as requiring that the
2609 charset have US-ASCII as a subset charset.

2610 4.1.8 `naturalLanguage'

2611 The `naturalLanguage' attribute syntax is a standard identifier for a natural language and optionally a
2612 country. The values for this syntax type are defined by RFC 1766 [RFC1766]. Though RFC 1766
2613 requires that the values be case-insensitive US-ASCII, IPP requires all lower case to simplify comparing
2614 by IPP clients and Printer objects. Examples include:

2615 `en': for English
2616 `en-us': for US English
2617 `fr': for French
2618 `de': for German
2619

2620 The maximum length of `naturalLanguage' values used to represent IPP attribute values is 63 octets.

2621 4.1.9 `mimeType'

2622 The `mimeType' attribute syntax is the Internet Media Type (sometimes called MIME type) as
2623 defined by RFC 2046 [RFC2046] and registered according to the procedures of RFC 2048 [RFC2048]
2624 for identifying a document format. The value MAY include a charset parameter, depending on the
2625 specification of the Media Type in the IANA Registry [IANA-MT]. Although most other IPP syntax
2626 types allow for only lower-cased values, this syntax type allows for mixed-case values which are case-
2627 insensitive.

2628 Examples are:

2629 `text/html': An HTML document
2630 `text/plain': A plain text document in US-ASCII (RFC 2046 indicates that in the absence of the
2631 charset parameter MUST mean US-ASCII rather than simply unspecified) [RFC2046].
2632 `text/plain; charset=US-ASCII': A plain text document in US-ASCII [52, 56].
2633 `text/plain; charset=ISO-8859-1': A plain text document in ISO 8859-1 (Latin 1) [ISO8859-1].
2634 `text/plain; charset=utf-8': A plain text document in ISO 10646 represented as UTF-8 [RFC2279]
2635 `application/postscript': A PostScript document [RFC2046]
2636 `application/vnd.hp-PCL': A PCL document [IANA-MT] (charset escape sequence embedded in the
2637 document data)

2638 'application/pdf': Portable Document Format - see IANA MIME Media Type registry
2639 'application/octet-stream': ~~(REQUIRED)~~ Auto-sense - see below

2640
2641

2642 4.1.9.1 Application/octet-stream -- Auto-Sensing the document format

2643 One special type is 'application/octet-stream'. If the Printer object supports this value, the Printer object
2644 MUST be capable of auto-sensing the format of the document data, either as part of the create operation
2645 and/or at document processing time. During auto-sensing, a Printer may determine that the document-
2646 data has a format that the Printer doesn't recognize. If the Printer determines this problem before
2647 returning an operation response-, it rejects the request and returns the 'client-error-document-format-not-
2648 supported' status code. If the Printer determines this problem after accepting the request and returning
2649 an operation response with one of the successful status codes, the Printer adds the 'unsupported-
2650 document-format' value to the job's "job-state-reasons" attribute. Issue 9 and Issue 10

2651 If the Printer object's default value attribute "document-format-default" is set to 'application/octet-
2652 stream', the Printer object not only supports auto-sensing of the document format, but will depend on the
2653 result of applying its auto-sensing when the client does not supply the "document-format" attribute. If
2654 the client supplies a document format value, the Printer MUST rely on the supplied attribute, rather than
2655 trust its auto-sensing algorithm. To summarize:

- 2656 1. If the client does not supply a document format value, the Printer MUST rely on its default value
2657 setting (which may be 'application/octet-stream' indicating an auto-sensing mechanism).
- 2658 2. If the client supplies a value other than 'application/octet-stream', the client is supplying valid
2659 information about the format of the document data and the Printer object MUST trust the client
2660 supplied value more than the outcome of applying an automatic format detection mechanism.
2661 For example, the client may be requesting the printing of a PostScript file as a 'text/plain'
2662 document. The Printer object MUST print a text representation of the PostScript commands
2663 rather than interpret the stream of PostScript commands and print the result.
- 2664 3. If the client supplies a value of 'application/octet-stream', the client is indicating that the Printer
2665 object MUST use its auto-sensing mechanism on the client supplied document data whether
2666 auto-sensing is the Printer object's default or not.

2667

2668 Note: Since the auto-sensing algorithm is probabilistic, if the client requests both auto-sensing
2669 ("document-format" set to 'application/octet-stream') and true fidelity ("ipp-attribute-fidelity" set to
2670 'true'), the Printer object might not be able to guarantee exactly what the end user intended (the auto-
2671 sensing algorithm might mistake one document format for another), but it is able to guarantee that its
2672 auto-sensing mechanism be used.

2673 The maximum length of a 'mimeType' value to represent IPP attribute values is 255 octets.

2674 4.1.10 'octetString'

2675 The 'octetString' attribute syntax is a sequence of octets encoded in a maximum of 1023 octets which is
2676 indicated in sub-section headers using the notation: octetString(MAX). This syntax type is used for
2677 opaque data.

2678 4.1.11 'boolean'

2679 The 'boolean' attribute syntax has only two values: 'true' and 'false'.

2680 4.1.12 'integer'

2681 The 'integer' attribute syntax is an integer value that is in the range from -2^{31} (MIN) to $2^{31} - 1$
2682 (MAX). Each individual attribute may specify the range constraint explicitly in sub-section headers if
2683 the range is different from the full range of possible integer values. For example: job-priority
2684 (integer(1:100)) for the "job-priority" attribute. However, the enforcement of that additional constraint is
2685 up to the IPP objects, not the protocol.

2686 4.1.13 'rangeOfInteger'

2687 The 'rangeOfInteger' attribute syntax is an ordered pair of integers that defines an inclusive range of
2688 integer values. The first integer specifies the lower bound and the second specifies the upper bound. If a
2689 range constraint is specified in the header description for an attribute in this document whose attribute
2690 syntax is 'rangeOfInteger' (i.e., 'X:Y' indicating X as a minimum value and Y as a maximum value), then
2691 the constraint applies to both integers.

2692 4.1.14 'dateTime'

2693 The 'dateTime' attribute syntax is a standard, fixed length, 11 octet representation of the "DateAndTime"
2694 syntax as defined in RFC 1903 [RFC1903]. RFC 1903 also identifies an 8 octet representation of a
2695 "DateAndTime" value, but IPP objects MUST use the 11 octet representation. A user interface will
2696 provide a mapping between protocol dateTime values and displayable user-friendly words or
2697 presentation values and phrases which are localized to the natural language and date format of the user.

2698 4.1.15 'resolution'

2699 The 'resolution' attribute syntax specifies a two-dimensional resolution in the indicated units. It consists
2700 of 3 values: a cross feed direction resolution (positive integer value), a feed direction resolution (positive
2701 integer value), and a units value. The semantics of these three components are taken from the Printer
2702 MIB [RFC1759] suggested values. That is, the cross feed direction component resolution component is
2703 the same as the prtMarkerAddressabilityXFeedDir object in the Printer MIB, the feed direction
2704 component resolution component is the same as the prtMarkerAddressabilityFeedDir in the Printer MIB,
2705 and the units component is the same as the prtMarkerAddressabilityUnit object in the Printer MIB
2706 (namely, '3' indicates dots per inch and '4' indicates dots per centimeter). All three values MUST be

2707 present even if the first two values are the same. Example: '300', '600', '3' indicates a 300 dpi cross-feed
2708 direction resolution, a 600 dpi feed direction resolution, since a '3' indicates dots per inch (dpi).

2709 4.1.16 '1setOf X'

2710 The '1setOf X' attribute syntax is 1 or more values of attribute syntax type X. This syntax type is used
2711 for multi-valued attributes. The syntax type is called '1setOf' rather than just 'setOf' as a reminder that
2712 the set of values MUST NOT be empty (i.e., a set of size 0). Sets are normally unordered. However
2713 each attribute description of this type may specify that the values MUST be in a certain order for that
2714 attribute.

2715 4.2 Job Template Attributes

2716 Job Template attributes describe job processing behavior. Support for Job Template attributes by a
2717 Printer object is OPTIONAL (see section 12.2.3 for a description of support for OPTIONAL attributes).
2718 Also, clients OPTIONALLY supply Job Template attributes in create requests.

2719 Job Template attributes conform to the following rules. For each Job Template attribute called "xxx":

2720 1. If the Printer object supports "xxx" then it MUST support both a "xxx-default" attribute (unless
2721 there is a "No" in the table below) and a "xxx-supported" attribute. If the Printer object doesn't
2722 support "xxx", then it MUST support neither an "xxx-default" attribute nor an "xxx-supported"
2723 attribute, and it MUST treat an attribute "xxx" supplied by a client as unsupported. An attribute
2724 "xxx" may be supported for some document formats and not supported for other document
2725 formats. For example, it is expected that a Printer object would only support "orientation-
2726 requested" for some document formats (such as 'text/plain' or 'text/html') but not others (such as
2727 'application/postscript').

2728
2729 2. "xxx" is OPTIONALLY supplied by the client in a create request. If "xxx" is supplied, the client
2730 is indicating a desired job processing behavior for this Job. When "xxx" is not supplied, the
2731 client is indicating that the Printer object apply its default job processing behavior at job
2732 processing time if the document content does not contain an embedded instruction indicating an
2733 xxx-related behavior.

2734
2735 Note: Since an administrator MAY change the default value attribute after a Job object has been
2736 submitted but before it has been processed, the default value used by the Printer object at job
2737 processing time may be different that the default value in effect at job submission time.

2738
2739 3. The "xxx-supported" attribute is a Printer object attribute that describes which job processing
2740 behaviors are supported by that Printer object. A client can query the Printer object to find out
2741 what xxx-related behaviors are supported by inspecting the returned values of the "xxx-
2742 supported" attribute.

2743
2744 Note: The "xxx" in each "xxx-supported" attribute name is singular, even though an "xxx-
2745 supported" attribute usually has more than one value, such as "job-sheet-supported", unless the

2746 "xxx" Job Template attribute is plural, such as "finishings" or "sides". In such cases the "xxx-
2747 supported" attribute names are: "finishings-supported" and "sides-supported".

2748

2749 4. The "xxx-default" default value attribute describes what will be done at job processing time when
2750 no other job processing information is supplied by the client (either explicitly as an IPP attribute
2751 in the create request or implicitly as an embedded instruction within the document data).

2752

2753 If an application wishes to present an end user with a list of supported values from which to choose, the
2754 application SHOULD query the Printer object for its supported value attributes. The application
2755 SHOULD also query the default value attributes. If the application then limits selectable values to only
2756 those value that are supported, the application can guarantee that the values supplied by the client in the
2757 create request all fall within the set of supported values at the Printer. When querying the Printer, the
2758 client MAY enumerate each attribute by name in the Get-Printer-Attributes Request, or the client MAY
2759 just name the "job-template" group in order to get the complete set of supported attributes (both
2760 supported and default attributes).

2761 The "finishings" attribute is an example of a Job Template attribute. It can take on a set of values such
2762 as 'staple', 'punch', and/or 'cover'. A client can query the Printer object for the "finishings-supported"
2763 attribute and the "finishings-default" attribute. The supported attribute contains a set of supported
2764 values. The default value attribute contains the finishing value(s) that will be used for a new Job if the
2765 client does not supply a "finishings" attribute in the create request and the document data does not
2766 contain any corresponding finishing instructions. If the client does supply the "finishings" attribute in
2767 the create request, the IPP object validates the value or values to make sure that they are a subset of the
2768 supported values identified in the Printer object's "finishings-supported" attribute. See section
2769 [3.1.73.2.1.2](#).

2770 The table below summarizes the names and relationships for all Job Template attributes. The first
2771 column of the table (labeled "Job Attribute") shows the name and syntax for each Job Template attribute
2772 in the Job object. These are the attributes that can optionally be supplied by the client in a create request.
2773 The last two columns (labeled "Printer: Default Value Attribute" and "Printer: Supported Values
2774 Attribute") shows the name and syntax for each Job Template attribute in the Printer object (the default
2775 value attribute and the supported values attribute). A "No" in the table means the Printer MUST NOT
2776 support the attribute (that is, the attribute is simply not applicable). For brevity in the table, the 'text' and
2777 'name' entries do not show the maximum length for each attribute.

2778			
2779	Job Attribute	Printer: Default Value Attribute	Printer: Supported Values Attribute
2780			
2781			
2782	job-priority (integer 1:100)	job-priority-default (integer 1:100)	job-priority-supported (integer 1:100)
2783			
2784			
2785	job-hold-until (type3 keyword name)	job-hold-until- default (type3 keyword name)	job-hold-until- supported (1setOf (type3 keyword name))
2786			
2787			
2788			
2789			
2790	job-sheets (type3 keyword name)	job-sheets-default (type3 keyword name)	job-sheets-supported (1setOf (type3 keyword name))
2791			
2792			
2793			
2794	multiple-document- handling (type2 keyword)	multiple-document- handling-default (type2 keyword)	multiple-document- handling-supported (1setOf type2 keyword)
2795			
2796			
2797			
2798	copies (integer (1:MAX))	copies-default (integer (1:MAX))	copies-supported (rangeOfInteger (1:MAX))
2799			
2800			
2801			
2802	finishings (1setOf type2 enum)	finishings-default (1setOf type2 enum)	finishings-supported (1setOf type2 enum)
2803			
2804			
2805	page-ranges (1setOf rangeOfInteger (1:MAX))	No	page-ranges- supported (boolean)
2806			
2807			
2808			
2809			
2810	sides (type2 keyword)	sides-default (type2 keyword)	sides-supported (1setOf type2 keyword)
2811			
2812			
2813	number-up (integer (1:MAX))	number-up-default (integer (1:MAX))	number-up-supported (1setOf integer (1:MAX) rangeOfInteger (1:MAX))
2814			
2815			
2816			
2817			
2818			
2819	orientation- requested (type2 enum)	orientation-requested- default (type2 enum)	orientation-requested- supported (1setOf type2 enum)
2820			
2821			
2822			
2823	media (type3 keyword name)	media-default (type3 keyword name)	media-supported (1setOf (type3 keyword name))
2824			
2825			
2826			
2827			media-ready

2828			(1setOf (<u></u>
2829			type3 keyword name))
2830	+-----+-----+-----+		
2831	printer-resolution	printer-resolution-	printer-resolution-
2832	(resolution)	default	supported
2833		(resolution)	(1setOf resolution)
2834	+-----+-----+-----+		
2835	print-quality	print-quality-default	print-quality-
2836	(type2 enum)	(type2 enum)	supported
2837			(1setOf type2 enum)
2838	+-----+-----+-----+		

2839

2840

2841 4.2.1 job-priority (integer(1:100))

2842 This attribute specifies a priority for scheduling the Job. A higher value specifies a higher priority. The
 2843 value 1 indicates the lowest possible priority. The value 100 indicates the highest possible priority.
 2844 Among those jobs that are ready to print, a Printer MUST print all jobs with a priority value of n before
 2845 printing those with a priority value of n-1 for all n.

2846 If the Printer object supports this attribute, it MUST always support the full range from 1 to 100. No
 2847 administrative restrictions are permitted. This way an end-user can always make full use of the entire
 2848 range with any Printer object. If privileged jobs are implemented outside IPP/1.1, they MUST have
 2849 priorities higher than 100, rather than restricting the range available to end-users.

2850 If the client does not supply this attribute and this attribute is supported by the Printer object, the Printer
 2851 object MUST use the value of the Printer object's "job-priority-default" at job submission time (unlike
 2852 most Job Template attributes that are used if necessary at job processing time).

2853 The syntax for the "job-priority-supported" is also integer(1:100). This single integer value indicates the
 2854 number of priority levels supported. The Printer object MUST take the value supplied by the client and
 2855 map it to the closest integer in a sequence of n integers values that are evenly distributed over the range
 2856 from 1 to 100 using the formula:

2857
$$\text{roundToNearestInt}((100x+50)/n)$$

2858 where n is the value of "job-priority-supported" and x ranges from 0 through n-1.

2859 For example, if n=1 the sequence of values is 50; if n=2, the sequence of values is: 25 and 75; if n = 3,
 2860 the sequence of values is: 17, 50 and 83; if n = 10, the sequence of values is: 5, 15, 25, 35, 45, 55, 65,
 2861 75, 85, and 95; if n = 100, the sequence of values is: 1, 2, 3, ... 100.

2862 If the value of the Printer object's "job-priority-supported" is 10 and the client supplies values in the
 2863 range 1 to 10, the Printer object maps them to 5, in the range 11 to 20, the Printer object maps them to
 2864 15, etc.

2865 4.2.2 job-hold-until (type3 keyword | name (MAX))

2866 This attribute specifies the named time period during which the Job MUST become a candidate for
2867 printing.

2868 Standard keyword values for named time periods are:

2869 'no-hold': immediately, if there are not other reasons to hold the job

2870 'indefinite': - the job is held indefinitely, until a client performs a Release-Job (section 3.3.6)

2871 'day-time': during the day

2872 'evening': evening

2873 'night': night

2874 'weekend': weekend

2875 'second-shift': second-shift (after close of business)

2876 'third-shift': third-shift (after midnight)

2877

2878 An administrator MUST associate allowable print times with a named time period (by means outside the
2879 scope of this IPP/1.1 document). An administrator is encouraged to pick names that suggest the type of
2880 time period. An administrator MAY define additional values using the 'name' or 'keyword' attribute
2881 syntax, depending on implementation.

2882 If the value of this attribute specifies a time period that is in the future, the Printer MUST add the 'job-
2883 hold-until-specified' value to the job's "job-state-reasons" attribute, move the job to the 'pending-held'
2884 state, and MUST NOT schedule the job for printing until the specified time-period arrives. When the
2885 specified time period arrives, the Printer MUST remove the 'job-hold-until-specified' value from the
2886 job's "job-state-reason" attribute and, if there are no other job state reasons that keep the job in the
2887 'pending-held' state, the Printer MUST consider the job as a candidate for processing by moving the job
2888 to the 'pending' state.

2889 If this job attribute value is the named value 'no-hold', or the specified time period has already started,
2890 the job MUST be a candidate for processing immediately.

2891 If the client does not supply this attribute and this attribute is supported by the Printer object, the Printer
2892 object MUST use the value of the Printer object's "job-hold-until-default" at job submission time (unlike
2893 most Job Template attributes that are used if necessary at job processing time).

2894 4.2.3 job-sheets (type3 keyword | name(MAX))

2895 This attribute determines which job start/end sheet(s), if any, MUST be printed with a job.

2896 Standard keyword values are:

2897 'none': no job sheet is printed

2898 'standard': one or more site specific standard job sheets are printed, e.g. a single start sheet or both
2899 start and end sheet is printed

2900

2901 An administrator MAY define additional values using the 'name' or 'keyword' attribute syntax, depending
2902 on implementation.

2903 Note: The effect of this attribute on jobs with multiple documents MAY be affected by the "multiple-
2904 document-handling" job attribute (section 4.2.4), depending on the job sheet semantics.

2905 4.2.4 multiple-document-handling (type2 keyword)

2906 This attribute is relevant only if a job consists of two or more documents. **This attribute MUST be**
2907 **supported if the Printer supports multiple documents per job (see sections 3.2.4 and 3.3.1). Issue 34** The
2908 attribute controls finishing operations and the placement of one or more print-stream pages into
2909 impressions and onto media sheets. When the value of the "copies" attribute exceeds 1, it also controls
2910 the order in which the copies that result from processing the documents are produced. For the purposes
2911 of this explanation, if "a" represents an instance of document data, then the result of processing the data
2912 in document "a" is a sequence of media sheets represented by "a(*)".

2913 Standard keyword values are:

2914 'single-document': If a Job object has multiple documents, say, the document data is called a and b,
2915 then the result of processing all the document data (a and then b) MUST be treated as a single
2916 sequence of media sheets for finishing operations; that is, finishing would be performed on the
2917 concatenation of the sequences a(*),b(*). The Printer object MUST NOT force the data in each
2918 document instance to be formatted onto a new print-stream page, nor to start a new impression
2919 on a new media sheet. If more than one copy is made, the ordering of the sets of media sheets
2920 resulting from processing the document data MUST be a(*), b(*), a(*), b(*), ..., and the Printer
2921 object MUST force each copy (a(*),b(*)) to start on a new media sheet.

2922 'separate-documents-uncollated-copies': If a Job object has multiple documents, say, the document
2923 data is called a and b, then the result of processing the data in each document instance MUST be
2924 treated as a single sequence of media sheets for finishing operations; that is, the sets a(*) and b(*)
2925 would each be finished separately. The Printer object MUST force each copy of the result of
2926 processing the data in a single document to start on a new media sheet. If more than one copy is
2927 made, the ordering of the sets of media sheets resulting from processing the document data
2928 MUST be a(*), a(*), ..., b(*), b(*)

2929 'separate-documents-collated-copies': If a Job object has multiple documents, say, the document data
2930 is called a and b, then the result of processing the data in each document instance MUST be
2931 treated as a single sequence of media sheets for finishing operations; that is, the sets a(*) and b(*)
2932 would each be finished separately. The Printer object MUST force each copy of the result of
2933 processing the data in a single document to start on a new media sheet. If more than one copy is
2934 made, the ordering of the sets of media sheets resulting from processing the document data
2935 MUST be a(*), b(*), a(*), b(*),

2936 'single-document-new-sheet': Same as 'single-document', except that the Printer object MUST
2937 ensure that the first impression of each document instance in the job is placed on a new media
2938 sheet. This value allows multiple documents to be stapled together with a single staple where
2939 each document starts on a new sheet.

2940

2941 The 'single-document' value is the same as 'separate-documents-collated-copies' with respect to ordering
 2942 of print-stream pages, but not media sheet generation, since 'single-document' will put the first page of
 2943 the next document on the back side of a sheet if an odd number of pages have been produced so far for
 2944 the job, while 'separate-documents-collated-copies' always forces the next document or document copy
 2945 on to a new sheet. In addition, if the "finishings" attribute specifies 'staple', then with 'single-document',
 2946 documents a and b are stapled together as a single document with no regard to new sheets, with 'single-
 2947 document-new-sheet', documents a and b are stapled together as a single document, but document b
 2948 starts on a new sheet, but with 'separate-documents-uncollated-copies' and 'separate-documents-collated-
 2949 copies', documents a and b are stapled separately.

2950 Note: None of these values provide means to produce uncollated sheets within a document, i.e., where
 2951 multiple copies of sheet n are produced before sheet n+1 of the same document.

2952 The relationship of this attribute and the other attributes that control document processing is described in
 2953 section 15.3.

2954 4.2.5 copies (integer(1:MAX))

2955 This attribute specifies the number of copies to be printed.

2956 On many devices the supported number of collated copies will be limited by the number of physical
 2957 output bins on the device, and may be different from the number of uncollated copies which can be
 2958 supported.

2959 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-
 2960 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other
 2961 attributes that control document processing is described in section 15.3.

2962 4.2.6 finishings (1setOf type2 enum)

2963 This attribute identifies the finishing operations that the Printer uses for each copy of each printed
 2964 document in the Job. For Jobs with multiple documents, the "multiple-document-handling" attribute
 2965 determines what constitutes a "copy" for purposes of finishing.

2966 Standard enum values are:

2967	Value	Symbolic Name and Description
2968		
2969	'3'	'none': Perform no finishing
2970	'4'	'staple': Bind the document(s) with one or more staples. The exact number and placement
2971		of the staples is site-defined.
2972	'5'	'punch': This value indicates that holes are required in the finished document. The exact
2973		number and placement of the holes is site-defined The punch specification MAY
2974		be satisfied (in a site- and implementation-specific manner) either by
2975		drilling/punching, or by substituting pre-drilled media.

- 2976 '6' 'cover': This value is specified when it is desired to select a non-printed (or pre-printed)
2977 cover for the document. This does not supplant the specification of a printed cover
2978 (on cover stock medium) by the document itself.
- 2979 '7' 'bind': This value indicates that a binding is to be applied to the document; the type and
2980 placement of the binding is site-defined.
- 2981
- 2982 '8' 'saddle-stitch': Bind the document(s) with one or more staples (wire stitches) along the
2983 middle fold. The exact number and placement of the staples and the middle fold
2984 is implementation and/or site-defined.
- 2985 '9' 'edge-stitch': Bind the document(s) with one or more staples (wire stitches) along one
2986 edge. The exact number and placement of the staples is implementation and/or
2987 site-defined.
- 2988 '10'-'19' reserved for future generic finishing enum values.

2989 The following values are more specific; they indicate a corner or an edge as if the document were a
2990 portrait document (see below):

- 2991 '20' 'staple-top-left': Bind the document(s) with one or more staples in the top left corner.
- 2992 '21' 'staple-bottom-left': Bind the document(s) with one or more staples in the bottom left
2993 corner.
- 2994 '22' 'staple-top-right': Bind the document(s) with one or more staples in the top right corner.
- 2995 '23' 'staple-bottom-right': Bind the document(s) with one or more staples in the bottom right
2996 corner.
- 2997 '24' 'edge-stitch-left': Bind the document(s) with one or more staples (wire stitches) along the
2998 left edge. The exact number and placement of the staples is implementation
2999 and/or site-defined.
- 3000 '25' 'edge-stitch-top': Bind the document(s) with one or more staples (wire stitches) along the
3001 top edge. The exact number and placement of the staples is implementation
3002 and/or site-defined.
- 3003 '26' 'edge-stitch-right': Bind the document(s) with one or more staples (wire stitches) along
3004 the right edge. The exact number and placement of the staples is implementation
3005 and/or site-defined.
- 3006 '27' 'edge-stitch-bottom': Bind the document(s) with one or more staples (wire stitches) along
3007 the bottom edge. The exact number and placement of the staples is
3008 implementation and/or site-defined.
- 3009 '28' 'staple-dual-left': Bind the document(s) with two staples (wire stitches) along the left
3010 edge assuming a portrait document (see above).
- 3011 '29' 'staple-dual-top': Bind the document(s) with two staples (wire stitches) along the top
3012 edge assuming a portrait document (see above).
- 3013 '30' 'staple-dual-right': Bind the document(s) with two staples (wire stitches) along the right
3014 edge assuming a portrait document (see above).
- 3015 '31' 'staple-dual-bottom': Bind the document(s) with two staples (wire stitches) along the
3016 bottom edge assuming a portrait document (see above).

3017 The 'staple-xxx' values are specified with respect to the document as if the document were a portrait
3018 document. If the document is actually a landscape or a reverse-landscape document, the client supplies
3019 the appropriate transformed value. For example, to position a staple in the upper left hand corner of a

3020 landscape document when held for reading, the client supplies the 'staple-bottom-left' value (since
3021 landscape is defined as a +90 degree rotation from portrait, i.e., anti-clockwise). On the other hand, to
3022 position a staple in the upper left hand corner of a reverse-landscape document when held for reading,
3023 the client supplies the 'staple-top-right' value (since reverse-landscape is defined as a -90 degree rotation
3024 from portrait, i.e., clockwise).

3025 The angle (vertical, horizontal, angled) of each staple with respect to the document depends on the
3026 implementation which may in turn depend on the value of the attribute.

3027 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-
3028 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other
3029 attributes that control document processing is described in section 15.3.

3030 If the client supplies a value of 'none' along with any other combination of values, it is the same as if
3031 only that other combination of values had been supplied (that is the 'none' value has no effect).

3032 4.2.7 page-ranges (1setOf rangeOfInteger (1:MAX))

3033 This attribute identifies the range(s) of print-stream pages that the Printer object uses for each copy of
3034 each document which are to be printed. Nothing is printed for any pages identified that do not exist in
3035 the document(s). Ranges MUST be in ascending order, for example: 1-3, 5-7, 15-19 and MUST NOT
3036 overlap, so that a non-spooling Printer object can process the job in a single pass. If the ranges are not
3037 ascending or are overlapping, the IPP object MUST reject the request and return the 'client-error-bad-
3038 request' status code. The attribute is associated with print-stream pages not application-numbered pages
3039 (for example, the page numbers found in the headers and or footers for certain word processing
3040 applications).

3041 For Jobs with multiple documents, the "multiple-document-handling" attribute determines what
3042 constitutes a "copy" for purposes of the specified page range(s). When "multiple-document-handling" is
3043 'single-document', the Printer object MUST apply each supplied page range once to the concatenation of
3044 the print-stream pages. For example, if there are 8 documents of 10 pages each, the page-range '41:60'
3045 prints the pages in the 5th and 6th documents as a single document and none of the pages of the other
3046 documents are printed. When "multiple-document-handling" is 'separate-documents_s-uncollated-copies'
3047 or 'separate-documents_s-collated-copies', the Printer object MUST apply each supplied page range
3048 repeatedly to each document copy. For the same job, the page-range '1:3, 10:10' would print the first 3
3049 pages and the 10th page of each of the 8 documents in the Job, as 8 separate documents.

3050 In most cases, the exact pages to be printed will be generated by a device driver and this attribute would
3051 not be required. However, when printing an archived document which has already been formatted, the
3052 end user may elect to print just a subset of the pages contained in the document. In this case, if page-
3053 range = n.m is specified, the first page to be printed will be page n. All subsequent pages of the
3054 document will be printed through and including page m.

3055 "page-ranges-supported" is a boolean value indicating whether or not the printer is capable of supporting
3056 the printing of page ranges. This capability may differ from one PDL to another. There is no "page-

3057 ranges-default" attribute. If the "page-ranges" attribute is not supplied by the client, all pages of the
3058 document will be printed.

3059 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-
3060 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other
3061 attributes that control document processing is described in section 15.3.

3062 4.2.8 sides (type2 keyword)

3063 This attribute specifies how print-stream pages are to be imposed upon the sides of an instance of a
3064 selected medium, i.e., an impression.

3065 The standard keyword values are:

3066 'one-sided': imposes each consecutive print-stream page upon the same side of consecutive media
3067 sheets.

3068 'two-sided-long-edge': imposes each consecutive pair of print-stream pages upon front and back sides
3069 of consecutive media sheets, such that the orientation of each pair of print-stream pages on the
3070 medium would be correct for the reader as if for binding on the long edge. This imposition is
3071 sometimes called 'duplex' or 'head-to-head'.

3072 'two-sided-short-edge': imposes each consecutive pair of print-stream pages upon front and back
3073 sides of consecutive media sheets, such that the orientation of each pair of print-stream pages on
3074 the medium would be correct for the reader as if for binding on the short edge. This imposition
3075 is sometimes called 'tumble' or 'head-to-toe'.

3076

3077 'two-sided-long-edge', 'two-sided-short-edge', 'tumble', and 'duplex' all work the same for portrait or
3078 landscape. However 'head-to-toe' is 'tumble' in portrait but 'duplex' in landscape. 'head-to-head' also
3079 switches between 'duplex' and 'tumble' when using portrait and landscape modes.

3080 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-
3081 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other
3082 attributes that control document processing is described in section 15.3.

3083 4.2.9 number-up (integer(1:MAX))

3084 This attribute specifies the number of print-stream pages to impose upon a single side of an instance of a
3085 selected medium. For example, if the value is:

3086 Value	Description
3087 '1'	the Printer MUST place one print-stream page on a single side of an instance of the 3088 selected medium (MAY add some sort of translation, scaling, or rotation).
3089 '2'	the Printer MUST place two print-stream pages on a single side of an instance of the 3090 selected medium (MAY add some sort of translation, scaling, or rotation).
3091 '4'	the Printer MUST place four print-stream pages on a single side of an instance of the 3092 selected medium (MAY add some sort of translation, scaling, or rotation).
3093	

3094

3095 This attribute primarily controls the translation, scaling and rotation of print-stream pages.

3096 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-
3097 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other
3098 attributes that control document processing is described in section 15.3.

3099 4.2.10 orientation-requested (type2 enum)

3100 This attribute indicates the desired orientation for printed print-stream pages; it does not describe the
3101 orientation of the client-supplied print-stream pages.

3102 For some document formats (such as 'application/postscript'), the desired orientation of the print-stream
3103 pages is specified within the document data. This information is generated by a device driver prior to
3104 the submission of the print job. Other document formats (such as 'text/plain') do not include the notion
3105 of desired orientation within the document data. In the latter case it is possible for the Printer object to
3106 bind the desired orientation to the document data after it has been submitted. It is expected that a Printer
3107 object would only support "orientations-requested" for some document formats (e.g., 'text/plain' or
3108 'text/html') but not others (e.g., 'application/postscript'). This is no different than any other Job Template
3109 attribute since section 4.2, item 1, points out that a Printer object may support or not support any Job
3110 Template attribute based on the document format supplied by the client. However, a special mention is
3111 made here since it is very likely that a Printer object will support "orientation-requested" for only a
3112 subset of the supported document formats.

3113 Standard enum values are:

3114 Value	Symbolic Name and Description
------------	-------------------------------

3115

3116 '3'	'portrait': The content will be imaged across the short edge of the medium.
----------	---

3117 '4'	'landscape': The content will be imaged across the long edge of the medium. Landscape 3118 is defined to be a rotation of the print-stream page to be imaged by +90 degrees 3119 with respect to the medium (i.e. anti-clockwise) from the portrait orientation. 3120 Note: The +90 direction was chosen because simple finishing on the long edge is 3121 the same edge whether portrait or landscape
----------	--

3122 '5'	'reverse-landscape': The content will be imaged across the long edge of the medium.
----------	---

3123	Reverse-landscape is defined to be a rotation of the print-stream page to be 3124 imaged by -90 degrees with respect to the medium (i.e. clockwise) from the 3125 portrait orientation. Note: The 'reverse-landscape' value was added because some 3126 applications rotate landscape -90 degrees from portrait, rather than +90 degrees.
------	--

3127 '6'	'reverse-portrait': The content will be imaged across the short edge of the medium.
----------	---

3128	Reverse-portrait is defined to be a rotation of the print-stream page to be imaged 3129 by 180 degrees with respect to the medium from the portrait orientation. Note: 3130 The 'reverse-portrait' value was added for use with the "finishings" attribute in 3131 cases where the opposite edge is desired for finishing a portrait document on 3132 simple finishing devices that have only one finishing position. Thus a 'text/plain'
------	---

3133 portrait document can be stapled "on the right" by a simple finishing device as is
3134 common use with some middle eastern languages such as Hebrew.
3135

3136 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-
3137 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other
3138 attributes that control document processing is described in section 15.3.

3139 4.2.11 media (type3 keyword | name(MAX))

3140 This attribute identifies the medium that the Printer uses for all impressions of the Job.

3141 The values for "media" include medium-names, medium-sizes, input-trays and electronic forms so that
3142 one attribute specifies the media. If a Printer object supports a medium name as a value of this attribute,
3143 such a medium name implicitly selects an input-tray that contains the specified medium. If a Printer
3144 object supports a medium size as a value of this attribute, such a medium size implicitly selects a
3145 medium name that in turn implicitly selects an input-tray that contains the medium with the specified
3146 size. If a Printer object supports an input-tray as the value of this attribute, such an input-tray implicitly
3147 selects the medium that is in that input-tray at the time the job prints. This case includes manual-feed
3148 input-trays. If a Printer object supports an electronic form as the value of this attribute, such an
3149 electronic form implicitly selects a medium-name that in turn implicitly selects an input-tray that
3150 contains the medium specified by the electronic form. The electronic form also implicitly selects an
3151 image that the Printer MUST merge with the document data as its prints each page.

3152 Standard keyword values are (taken from ISO DPA and the Printer MIB) and are listed in section 14. An
3153 administrator MAY define additional values using the 'name' or 'keyword' attribute syntax, depending on
3154 implementation.

3155 There is also an additional Printer attribute named "media-ready" which differs from "media-supported"
3156 in that legal values only include the subset of "media-supported" values that are physically loaded and
3157 ready for printing with no operator intervention required. If an IPP object supports "media-supported", it
3158 NEED NOT support "media-ready".

3159 The relationship of this attribute and the other attributes that control document processing is described in
3160 section 15.3.

3161 4.2.12 printer-resolution (resolution)

3162 This attribute identifies the resolution that Printer uses for the Job.

3163 4.2.13 print-quality (type2 enum)

3164 This attribute specifies the print quality that the Printer uses for the Job.

3165 The standard enum values are:

3166 Value Symbolic Name and Description

3167
3168 '3' 'draft': lowest quality available on the printer
3169 '4' 'normal': normal or intermediate quality on the printer
3170 '5' 'high': highest quality available on the printer
3171

3172 4.3 Job Description Attributes

3173 The attributes in this section form the attribute group called "job-description". The following table
3174 summarizes these attributes. The third column indicates whether the attribute is a REQUIRED attribute
3175 that MUST be supported by Printer objects. If it is not indicated as REQUIRED, then it is OPTIONAL.
3176 The maximum size in octets for 'text' and 'name' attributes is indicated in parentheses.

3177	+	-----+	-----+	-----+
3178		Attribute	Syntax	REQUIRED?
3179	+	-----+	-----+	-----+
3180		job-uri	uri	REQUIRED
3181	+	-----+	-----+	-----+
3182		job-id	integer(1:MAX)	REQUIRED
3183	+	-----+	-----+	-----+
3184		job-printer-uri	uri	REQUIRED
3185	+	-----+	-----+	-----+
3186		job-more-info	uri	
3187	+	-----+	-----+	-----+
3188		job-name	name (MAX)	REQUIRED
3189	+	-----+	-----+	-----+
3190		job-originating-user-name	name (MAX)	REQUIRED
3191	+	-----+	-----+	-----+
3192		job-state	type1 enum	REQUIRED
3193	+	-----+	-----+	-----+
3194		job-state-reasons	1setOf type2 keyword	<u>REQUIRED</u>
3195	+	-----+	-----+	-----+
3196		job-state-message	text (MAX)	
3197	+	-----+	-----+	-----+
3198		number-of-documents	integer (0:MAX)	
3199	+	-----+	-----+	-----+
3200		output-device-assigned	name (127)	
3201	+	-----+	-----+	-----+
3202		time-at-creation	integer (0:MAX)	<u>REQUIRED</u>
3203	+	-----+	-----+	-----+
3204		time-at-processing	integer (0:MAX)	<u>REQUIRED</u>
3205	+	-----+	-----+	-----+
3206		time-at-completed	integer (0:MAX)	<u>REQUIRED</u>
3207	+	-----+	-----+	-----+
3208		<u>job-printer-up-time</u>	<u>integer (1:MAX)</u>	<u>REQUIRED</u>
3209	+	-----+	-----+	-----+
3210		number-of-intervening-jobs	integer (0:MAX)	
3211	+	-----+	-----+	-----+
3212		job-message-from-operator	text (127)	
3213	+	-----+	-----+	-----+
3214		job-k-octets	integer (0:MAX)	
3215	+	-----+	-----+	-----+
3216		job-impressions	integer (0:MAX)	
3217	+	-----+	-----+	-----+
3218		job-media-sheets	integer (0:MAX)	
3219	+	-----+	-----+	-----+
3220		job-k-octets-processed	integer (0:MAX)	
3221	+	-----+	-----+	-----+
3222		job-impressions-completed	integer (0:MAX)	
3223	+	-----+	-----+	-----+
3224		job-media-sheets-completed	integer (0:MAX)	
3225	+	-----+	-----+	-----+
3226		attributes-charset	charset	REQUIRED

3227	+-----+-----+-----+
3228	attributes-natural-language naturalLanguage REQUIRED
3229	+-----+-----+-----+
3230	
3231	

3232 4.3.1 job-uri (uri)

3233 This REQUIRED attribute contains the URI for the job. The Printer object, on receipt of a new job,
 3234 generates a URI which identifies the new Job. The Printer object returns the value of the "job-uri"
 3235 attribute as part of the response to a create request. The precise format of a Job URI is implementation
 3236 dependent. If the Printer object supports more than one URI and there is some relationship between the
 3237 newly formed Job URI and the Printer object's URI, the Printer object uses the Printer URI supplied by
 3238 the client in the create request. For example, if the create request comes in over a secure channel, the
 3239 new Job URI MUST use the same secure channel. This can be guaranteed because the Printer object is
 3240 responsible for generating the Job URI and the Printer object is aware of its security configuration and
 3241 policy as well as the Printer URI used in the create request.

3242 For a description of this attribute and its relationship to "job-id" and "job-printer-uri" attribute, see the
 3243 discussion in section 2.4 on "Object Identity".

3244 4.3.2 job-id (integer(1:MAX))

3245 This REQUIRED attribute contains the ID of the job. The Printer, on receipt of a new job, generates an
 3246 ID which identifies the new Job on that Printer. The Printer returns the value of the "job-id" attribute as
 3247 part of the response to a create request. The 0 value is not included to allow for compatibility with
 3248 SNMP index values which also cannot be 0.

3249 For a description of this attribute and its relationship to "job-uri" and "job-printer-uri" attribute, see the
 3250 discussion in section 2.4 on "Object Identity".

3251 4.3.3 job-printer-uri (uri)

3252 This REQUIRED attribute identifies the Printer object that created this Job object. When a Printer
 3253 object creates a Job object, it populates this attribute with the Printer object URI that was used in the
 3254 create request. This attribute permits a client to identify the Printer object that created this Job object
 3255 when only the Job object's URI is available to the client. The client queries the creating Printer object to
 3256 determine which languages, charsets, operations, are supported for this Job.

3257 For a description of this attribute and its relationship to "job-uri" and "job-id" attribute, see the
 3258 discussion in section 2.4 on "Object Identity".

3259 4.3.4 job-more-info (uri)

3260 Similar to "printer-more-info", this attribute contains the URI referencing some resource with more
 3261 information about this Job object, perhaps an HTML page containing information about the Job.

3262 4.3.5 job-name (name(MAX))

3263 This REQUIRED attribute is the name of the job. It is a name that is more user friendly than the "job-
 3264 uri" attribute value. It does not need to be unique between Jobs. The Job's "job-name" attribute is set to
 3265 the value supplied by the client in the "job-name" operation attribute in the create request (see Section
 3266 3.2.1.1). If, however, the "job-name" operation attribute is not supplied by the client in the create
 3267 request, the Printer object, on creation of the Job, MUST generate a name. The printer SHOULD
 3268 generate the value of the Job's "job-name" attribute from the first of the following sources that produces
 3269 a value: 1) the "document-name" operation attribute of the first (or only) document, 2) the "document-
 3270 URI" attribute of the first (or only) document, or 3) any other piece of Job specific and/or Document
 3271 Content information.

3272 4.3.6 job-originating-user-name (name(MAX))

3273 This REQUIRED attribute contains the name of the end user that submitted the print job. The Printer
 3274 object sets this attribute to the most authenticated printable name that it can obtain from the
 3275 authentication service over which the IPP operation was received. Only if such is not available, does the
 3276 Printer object use the value supplied by the client in the "requesting-user-name" operation attribute of the
 3277 create operation (see Section 8).

3278 Note: The Printer object needs to keep an internal originating user id of some form, typically as a
 3279 credential of a principal, with the Job object. Since such an internal attribute is implementation-
 3280 dependent and not of interest to clients, it is not specified as a Job Description attribute. This originating
 3281 user id is used for authorization checks (if any) on all subsequent operation.

3282 4.3.7 job-state (type1 enum)

3283 This REQUIRED attribute identifies the current state of the job. Even though the IPP protocol defines
 3284 seven values for job states (plus the out-of-band 'unknown' value - see Section 4.1), implementations
 3285 only need to support those states which are appropriate for the particular implementation. In other
 3286 words, a Printer supports only those job states implemented by the output device and available to the
 3287 Printer object implementation.

3288 Standard enum values are:

3289	Values	Symbolic Name and Description
3290		
3291	'3'	'pending': The job is a candidate to start processing, but is not yet processing.
3292		
3293	'4'	'pending-held': The job is not a candidate for processing for any number of reasons but 3294 will return to the 'pending' state as soon as the reasons are no longer present. The 3295 job's "job-state-reason" attribute MUST indicate why the job is no longer a 3296 candidate for processing.
3297		
3298	'5'	'processing': One or more of:
3299		

- 3300 1. the job is using, or is attempting to use, one or more purely software processes
 3301 that are analyzing, creating, or interpreting a PDL, etc.,
 3302 2. the job is using, or is attempting to use, one or more hardware devices that are
 3303 interpreting a PDL, making marks on a medium, and/or performing finishing,
 3304 such as stapling, etc.,
 3305 3. the Printer object has made the job ready for printing, but the output device is
 3306 not yet printing it, either because the job hasn't reached the output device or
 3307 because the job is queued in the output device or some other spooler, awaiting the
 3308 output device to print it.
 3309

3310 When the job is in the 'processing' state, the entire job state includes the detailed
 3311 status represented in the Printer **object's** "printer-state", "printer-state-reasons",
 3312 and "printer-state-message" attributes.

3313 Implementations MAY, though they NEED NOT, include additional values in the
 3314 job's "job-state-reasons" attribute to indicate the progress of the job, such as
 3315 adding the 'job-printing' value to indicate when the output device is actually
 3316 making marks on paper and/or the 'processing-to-stop-point' value to indicate that
 3317 the IPP object is in the process of canceling or aborting the job. Most
 3318 implementations won't bother with this nuance.
 3319

3320 '6' 'processing-stopped': The job has stopped while processing for any number of reasons
 3321 and will return to the 'processing' state as soon as the reasons are no longer
 3322 present.
 3323

3324 The job's "job-state-reason" attribute MAY indicate why the job has stopped
 3325 processing. For example, if the output device is stopped, the 'printer-stopped'
 3326 value MAY be included in the job's "job-state-reasons" attribute.
 3327

3328 Note: When an output device is stopped, the device usually indicates its condition
 3329 in human readable form locally at the device. A client can obtain more complete
 3330 device status remotely by querying the Printer object's "printer-state", "printer-
 3331 state-reasons" and "printer-state-message" attributes.
 3332

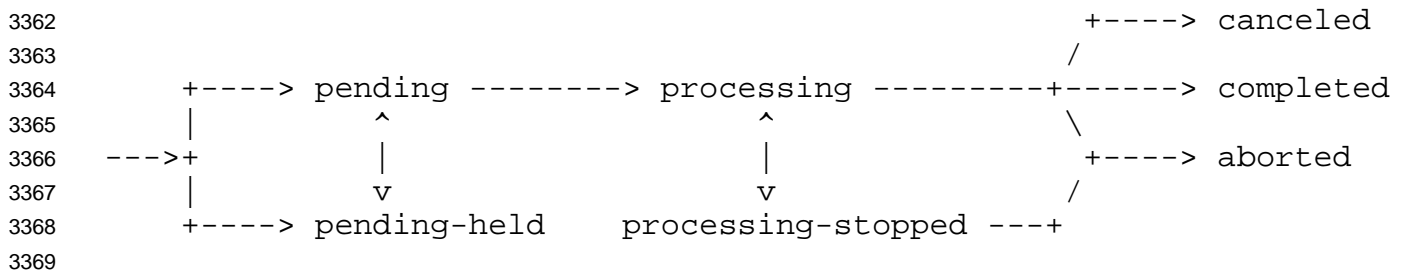
3333 '7' 'canceled': The job has been canceled by a Cancel-Job operation and the Printer object
 3334 has completed canceling the job and all job status attributes have reached their
 3335 final values for the job. While the Printer object is canceling the job, the job
 3336 remains in its current state, but the job's "job-state-reasons" attribute SHOULD
 3337 contain the 'processing-to-stop-point' value and one of the 'canceled-by-user',
 3338 'canceled-by-operator', or 'canceled-at-device' value. When the job moves to the
 3339 'canceled' state, the 'processing-to-stop-point' value, if present, MUST be
 3340 removed, but the 'canceled-by-xxx', if present, MUST remain.
 3341

3342 '8' 'aborted': The job has been aborted by the system, usually while the job was in the
 3343 'processing' or 'processing-stopped' state and the Printer has completed aborting
 3344 the job and all job status attributes have reached their final values for the job.
 3345 While the Printer object is aborting the job, the job remains in its current state, but
 3346 the job's "job-state-reasons" attribute SHOULD contain the 'processing-to-stop-
 3347 point' and 'aborted-by-system' values. When the job moves to the 'aborted' state,
 3348 the 'processing-to-stop-point' value, if present, MUST be removed, but the
 3349 'aborted-by-system' value, if present, MUST remain.

3351 '9' 'completed': The job has completed successfully or with warnings or errors after
 3352 processing and all of the job media sheets have been successfully stacked in the
 3353 appropriate output bin(s) and all job status attributes have reached their final
 3354 values for the job. The job's "job-state-reasons" attribute SHOULD contain one
 3355 of: 'completed-successfully', 'completed-with-warnings', or 'completed-with-errors'
 3356 values.

3358 The final value for this attribute MUST be one of: 'completed', 'canceled', or 'aborted' before the Printer
 3359 removes the job altogether. The length of time that jobs remain in the 'canceled', 'aborted', and
 3360 'completed' states depends on implementation. See section 4.3.7.1.

3361 The following figure shows the normal job state transitions.



3370 Normally a job progresses from left to right. Other state transitions are unlikely, but are not forbidden.
 3371 Not shown are the transitions to the 'canceled' state from the 'pending', 'pending-held', and 'processing-
 3372 stopped' states.

3373 Jobs reach one of the three terminal states: 'completed', 'canceled', or 'aborted', after the jobs have
 3374 completed all activity, including stacking output media, after the jobs have completed all activity, and all
 3375 job status attributes have reached their final values for the job.

3376 Note: As with all other IPP attributes, if the implementation can-not determine the correct value for this
 3377 attribute, it SHOULD respond with the out-of-band value 'unknown' (see section 4.1) rather than try to
 3378 guess at some possibly incorrect value and give the end user the wrong impression about the state of the
 3379 Job object. For example, if the implementation is just a gateway into some printing system from which
 3380 it can normally get status, but temporarily is unable, then the implementation should return the
 3381 'unknown' value. However, if the implementation is a gateway to a printing system that does not never
 3382 provides detailed status about the print job, the implementation MAY set the IPP Job object's state might

3383 ~~literally be 'unknown' to 'completed', provided that it also sets the 'queued-in-device' value in the job's~~
3384 ~~"job-state-reasons" attribute (see section 4.3.8). Issue 14~~

3385 4.3.7.1 Partitioning of Job States

3386 This section partitions the 7 job states into phases: Job Not Completed, Job Retention, Job History, and
3387 Job Removal. This section also explains the 'job-restartable' value of the "job-state-reasons" Job
3388 Description attribute for use with the Restart-Job operation.

3389 Job Not Completed: When a job is in the 'pending', 'pending-held', 'processing', or 'processing-stopped'
3390 states, the job is not completed.

3391 Job Retention: When a job enters one of the three terminal job states: 'completed', 'canceled', or
3392 'aborted', the IPP Printer object MAY "retain" the job in a restartable condition for an implementation-
3393 defined time period. This time period MAY be zero seconds and MAY depend on the terminal job state.
3394 This phase is called Job Retention. While in the Job Retention phase, the job's document data is retained
3395 and a client may restart the job using the Restart-Job operation. If the IPP object supports ~~the "job-state-~~
3396 ~~reasons" attribute and~~ Issue 30 the Restart-Job operation, then it SHOULD indicate that the job is
3397 restartable by adding the 'job-restartable' value to the job's "job-state-reasons" attribute (see Section
3398 4.3.8) during the Job Retention phase.

3399 Job History: After the Job Retention phase expires for a job, the Printer object deletes the document
3400 data for the job and the job becomes part of the Job History. The Printer object MAY also delete any
3401 number of the job attributes. Since the job is no longer restartable, the Printer object MUST remove the
3402 'job-restartable' value from the job's "job-state-reasons" attribute, if present.

3403 Job Removal: After the job has remained in the Job History for an implementation-defined time, such as
3404 when the number of jobs exceeds a fixed number or after a fixed time period (which MAY be zero
3405 seconds), the IPP Printer removes the job from the system.

3406 Using the Get-Jobs operation and supplying the 'not-completed' value for the "which-jobs" operation
3407 attribute, a client is requesting jobs in the Job Not Completed phase. Using the Get-Jobs operation and
3408 supplying the 'completed' value for the "which-jobs" operation attribute, a client is requesting jobs in the
3409 Job Retention and Job History phases. Using the Get-Job-Attributes operation, a client is requesting a
3410 job in any phase except Job Removal. After Job Removal, the Get-Job-Attributes and Get-Jobs
3411 operations no longer are capable of returning any information about a job.

3412 4.3.8 job-state-reasons (1setOf type2 keyword)

3413 This **REQUIRED** attribute provides additional information about the job's current state, i.e., information
3414 that augments the value of the job's "job-state" attribute.

3415 ~~While implementation of this attribute is REQUIRED, implementation of these values is OPTIONAL,~~
3416 ~~i.e., a Printer NEED NOT implement them, even if (1) the output device supports the functionality~~
3417 ~~represented by the reason and (2) is available to the Printer object implementation. Issue 30~~ These
3418 values MAY be used with any job state or states for which the reason makes sense. Furthermore, when
3419 implemented, the Printer MUST return these values when the reason applies and MUST NOT return

3420 them when the reason no longer applies whether the value of the Job's "job-state" attribute changed or
3421 not. When the Job does not have any reasons for being in its current state, the value of the Job's "job-
3422 state-reasons" attribute MUST be 'none'.

3423 Note: While values cannot be added to the 'job-state' attribute without impacting deployed clients that
3424 take actions upon receiving "job-state" values, it is the intent that additional "job-state-reasons" values
3425 can be defined and registered without impacting such deployed clients. In other words, the "job-state-
3426 reasons" attribute is intended to be extensible.

3427 The following standard keyword values are defined. For ease of understanding, the values are presented
3428 in the order in which the reasons are likely to occur (if implemented), starting with the 'job-incoming'
3429 value:

3430 'none': There are no reasons for the job's current state. This state reason is semantically equivalent
3431 to "job-state-reasons" without any value and MUST be used when there is no other value, since
3432 the 1setOf attribute syntax requires at least one value.

3433 'job-incoming': The Create-Job operation has been accepted by the Printer, but the Printer is
3434 expecting additional Send-Document and/or Send-URI operations and/or is accessing/accepting
3435 document data.

3436 'job-data-insufficient': The Create-Job operation has been accepted by the Printer, but the Printer is
3437 expecting additional document data before it can move the job into the 'processing' state. If a
3438 Printer starts printing processing before it has received all data, the Printer removes the 'job-data-
3439 insufficient' reason, but the 'job-incoming' remains. If a Printer starts printing processing after it
3440 has received all data, the Printer removes the 'job-data-insufficient' reason and the 'job-incoming'
3441 at the same time. Issue 13

3442 'document-access-error': After accepting a Print-URI or Send-URI request, the Printer could not
3443 access one or more documents passed by reference. This reason is intended to cover any file
3444 access problem, including file does not exist and access denied because of an access control
3445 problem. Whether the Printer aborts the job and moves the job to the 'aborted' job state or prints
3446 all documents that are accessible and moves the job to the 'completed' job state and adds the
3447 'completed-with-errors' value in the job's "job-state-reasons" attribute depends on implementation
3448 and/or site policy. Issue 35

3449 'submission-interrupted': The job was not completely submitted for some unforeseen reason, such
3450 as: (1) the Printer has crashed before the job was closed by the client, (2) the Printer or the
3451 document transfer method has crashed in some non-recoverable way before the document data
3452 was entirely transferred to the Printer, (3) the client crashed or failed to close the job before the
3453 time-out period. See section 4.4.31.

3454 'job-outgoing': The Printer is transmitting the job to the output device.

3455 'job-hold-until-specified': The value of the job's "job-hold-until" attribute was specified with a time
3456 period that is still in the future. The job MUST NOT be a candidate for processing until this
3457 reason is removed and there are no other reasons to hold the job.

3458 'resources-are-not-ready': At least one of the resources needed by the job, such as media, fonts,
3459 resource objects, etc., is not ready on any of the physical printer's for which the job is a candidate.
3460 This condition MAY be detected when the job is accepted, or subsequently while the job is
3461 pending or processing, depending on implementation. The job may remain in its current state or
3462 be moved to the 'pending-held' state, depending on implementation and/or job scheduling policy.

- 3463 'printer-stopped-partly': The value of the Printer's "printer-state-reasons" attribute contains the value
3464 'stopped-partly'.
- 3465 'printer-stopped': The value of the Printer's "printer-state" attribute is 'stopped'.
- 3466 'job-interpreting': Job is in the 'processing' state, but more specifically, the Printer is interpreting the
3467 document data.
- 3468 'job-queued': Job is in the 'processing' state, but more specifically, the Printer has queued the
3469 document data.
- 3470 'job-transforming': Job is in the 'processing' state, but more specifically, the Printer is interpreting
3471 document data and producing another electronic representation.
- 3472 'job-queued-for-marker': Job is in any of the 'pending-held', 'pending', or 'processing' states, but more
3473 specifically, the Printer has completed enough processing of the document to be able to start
3474 marking and the job is waiting for the marker. Systems that require human intervention to
3475 release jobs using the Release-Job operation, put the job into the 'pending-held' job state.
3476 Systems that automatically select a job to use the marker put the job into the 'pending' job state
3477 or keep the job in the 'processing' job state while waiting for the marker, depending on
3478 implementation. All implementations put the job into (or back into) the 'processing' state when
3479 marking does begin. Issue 31
- 3480 'job-printing': The output device is marking media. This value is useful for Printers which spend a
3481 great deal of time processing (1) when no marking is happening and then want to show that
3482 marking is now happening or (2) when the job is in the process of being canceled or aborted
3483 while the job remains in the 'processing' state, but the marking has not yet stopped so that
3484 impression or sheet counts are still increasing for the job.
- 3485 'job-canceled-by-user': The job was canceled by the owner of the job using the Cancel-Job request,
3486 i.e., by a user whose authenticated identity is the same as the value of the originating user that
3487 created the Job object, or by some other authorized end-user, such as a member of the job
3488 owner's security group.
- 3489 'job-canceled-by-operator': The job was canceled by the operator using the Cancel-Job request, i.e.,
3490 by a user who has been authenticated as having operator privileges (whether local or remote). If
3491 the security policy is to allow anyone to cancel anyone's job, then this value may be used when
3492 the job is canceled by other than the owner of the job. For such a security policy, in effect,
3493 everyone is an operator as far as canceling jobs with IPP is concerned.
- 3494 'job-canceled-at-device': The job was canceled by an unidentified local user, i.e., a user at a console
3495 at the device.
- 3496 'aborted-by-system': The job (1) is in the process of being aborted, (2) has been aborted by the
3497 system and placed in the 'aborted' state, or (3) has been aborted by the system and placed in the
3498 'pending-held' state, so that a user or operator can manually try the job again.
- 3499 'unsupported-compression': The job was aborted by the system because the Printer determined while
3500 attempting to decompress the document-data's that the compression is actually not among those
3501 supported by the Printer. Issue 6
- 3502 'compression-error': The job was aborted by the system because the Printer encountered an error in
3503 the document-data while decompressing it. If the Printer posts this reason, the document-data has
3504 already passed any tests that would have led to the 'unsupported-compression' job-state-reason.
3505 Issue 6
- 3506 'unsupported-document-format': The job was aborted by the system because the document-data's
3507 document-format is not among those supported by the Printer. If the client specifies the

3508 document-format as 'application/octet-stream', the printer MAY abort the job and post this reason
 3509 even though the format is a member of the "document-format-supported" printer attribute, but
 3510 not among the auto-sensed document-formats. Issue 3
 3511 'document-format-error': The job was aborted by the system because the Printer encountered an error
 3512 in the document-data while processing it. If the Printer posts this reason, the document-data has
 3513 already passed any tests that would have led to the 'unsupported-document-format' job-state-
 3514 reason. Issue 3
 3515 'processing-to-stop-point': The requester has issued a Cancel-Jjob operation or the Printer object has
 3516 aborted the job, but is still performing some actions on the job until a specified stop point occurs
 3517 or job termination/cleanup is completed.
 3518

3519 This reason is recommended to be used in conjunction with the 'processing' job state to indicate
 3520 that the Printer object is still performing some actions on the job while the job remains in the
 3521 'processing' state. After all the job's job description attributes have stopped incrementing, the
 3522 Printer object moves the job from the 'processing' state to the 'canceled' or 'aborted' job states.
 3523

3524 'service-off-line': The Printer is off-line and accepting no jobs. All 'pending' jobs are put into the
 3525 'pending-held' state. This situation could be true if the service's or document transform's input is
 3526 impaired or broken.

3527 'job-completed-successfully': The job completed successfully.

3528 'job-completed-with-warnings': The job completed with warnings.

3529 'job-completed-with-errors': The job completed with errors (and possibly warnings too).

3530 'job-restartable' - This job is retained (see section 4.3.7.1) and is currently able to be restarted using
 3531 the Restart-Job operation (see section 3.3.7). If 'job-restartable' is a value of the job's 'job-state-

3532 reasons' attribute, then the IPP object MUST accept a Restart-Job operation for that job.
 3533 'queued-in-device': The job has been forwarded to a device or print system which that is unable to
 3534 send back status. The Printer sets the job's "job-state" attribute to 'completed' and adds the
 3535 'queued-in-device' value to the job's "job-state-reasons" attribute to indicate that the Printer has
 3536 no additional information about the job and never will have any better information. See note in
 3537 section 4.3.7. Issue 14

3538 4.3.9 job-state-message (text(MAX))

3539 This attribute specifies information about the "job-state" and "job-state-reasons" attributes in human
 3540 readable text. If the Printer object supports this attribute, the Printer object MUST be able to generate
 3541 this message in any of the natural languages identified by the Printer's "generated-natural-language-

3542 supported" attribute (see the "attributes-natural-language" operation attribute specified in Section
 3543 3.1.4.1).
 3544 Note: the value SHOULD NOT contain additional information not contained in the values of the "job-

3549 4.3.10 number-of-documents (integer(0:MAX))

3550 This attribute indicates the number of documents in the job, i.e., the number of Send-Document, Send-
3551 URI, Print-Job, or Print-URI operations that the Printer has accepted for this job, regardless of whether
3552 the document data has reached the Printer object or not.

3553 Implementations supporting the OPTIONAL Create-Job/Send-Document/Send-URI operations
3554 SHOULD support this attribute so that clients can query the number of documents in each job.

3555 4.3.11 output-device-assigned (name(127))

3556 This attribute identifies the output device to which the Printer object has assigned this job. If an output
3557 device implements an embedded Printer object, the Printer object NEED NOT set this attribute. If a
3558 print server implements a Printer object, the value MAY be empty (zero-length string) or not returned
3559 until the Printer object assigns an output device to the job. This attribute is particularly useful when a
3560 single Printer object support multiple devices (so called "fan-out").

3561 4.3.12 Event Time Job Description Attributes Issue 17

3562 This section defines the Job Description attributes that indicate the time at which certain events occur for
3563 a job. The attribute syntax MUST be either 'integer' or 'dateTime' for any response in which the
3564 "version-number" parameter is supplied as '1.1', but MUST be an 'integer' for any response in which the
3565 "version-number" parameter is supplied as '1.0', for compatibility with IPP/1.0 [RFC2566]. See section
3566 3.1.8.

3567 In order to populate these Event Time Job Description Attributes, the Printer object copies either:

- 3568 1. the value in its "printer-current-time" attribute for the 'dateTime' value at the time the event
3569 occurred if the printer supports the attribute "printer-current-time" and its value is not the out-
3570 of-band 'no-value' value,
- 3571 2. the value in its "printer-up-time" attribute for the 'integer' value at the time the event occurred
3572 otherwise

3573 Note: because the time MAY become known to the Printer some time after power-up, a client could
3574 receive jobs that contain some Event Time Job Description Attributes that use the 'integer' time tick
3575 representation while the later events use the 'dateTime' date/time representation.

3576 If the Printer implementation keeps jobs persistently across power cycles, then an implementation
3577 MUST reset its "printer-up-time" attribute to 1 on each power-up. In addition, an implementation that
3578 uses the 'integer' form MUST change all of its Event Time Job Description attributes for those persistent
3579 jobs either:

- 3580 1. to 0 to indicate that the event happened before the most recent power up

3581 2. to the negative of the number of seconds before the most recent power-up that the event took
3582 place, though the negative number NEED NOT reflect the exact number of seconds

3583 An implementation that uses the 'dateTime' form does not change the values of any of its Event Time
3584 Job Description Attributes for persistent jobs on power-up.

3585 ~~4.3.12.1~~ 4.3.12.1 time-at-creation (integer(MIN@:MAX) | dateTime)

3586 This **REQUIRED** attribute indicates the ~~point in~~ time at which the Job object was created. ~~In order to~~
3587 ~~populate this attribute, the Printer object uses the value in its "printer-up-time" attribute at the time the~~
3588 ~~Job object is created.~~

3589 ~~2.3.13~~ 4.3.12.2 time-at-processing (integer(MIN@:MAX) | dateTime)

3590 This **REQUIRED** attribute indicates the ~~point in~~ time at which the Job object began processing. The
3591 out-of-band 'no-value' value is returned if the job has not yet been in the 'processing' state (see the
3592 beginning of Section 4.1).~~In order to populate this attribute, the Printer object uses the value in its~~
3593 ~~"printer-up-time" attribute at the time the Job object is moved into the 'processing' state for the first time.~~

3595 ~~4.3.11.3~~ 4.3.12.3 time-at-completed (integer(MIN@:MAX) | dateTime)

3596 This **REQUIRED** attribute indicates the ~~point in~~ time at which the Job object completed (or was
3597 cancelled or aborted). The out-of-band 'no-value' value is returned if the job has not yet completed, been
3598 canceled, or aborted (see the beginning of Section 4.1).~~In order to populate this attribute, the Printer~~
3599 ~~object uses the value in its "printer-up-time" attribute at the time the Job object is moved into the~~
3600 ~~'completed' or 'canceled' or 'aborted' state.~~

3601 4.3.12.4 job-printer-up-time (integer(1:MAX)) **Issue 17**

3602 This REQUIRED Job Description attribute indicates the amount of time (in seconds) that the Printer
3603 implementation has been up and running. This attribute is an alias for the "printer-up-time" Printer
3604 Description attribute (see Section 4.4.29).

3605 Note: A client MAY request this attribute in a Get-Job-Attributes or Get-Jobs request and use the value
3606 returned in combination with other requested Event Time Job Description Attributes in order to display
3607 time attributes to a user when the IPP Printer returns them using the 'integer' attribute syntax. The
3608 difference between this attribute and the 'integer' value of a "time-at-xxx" attribute is the number of
3609 seconds ago that the "time-at-xxx" event occurred. A client can compute the wall-clock time at which
3610 the "time-at-xxx" event occurred by subtracting this difference from the client's wall-clock time.

3611 4.3.13 number-of-intervening-jobs (integer(0:MAX))

3612 This attribute indicates the number of jobs that are "ahead" of this job in the relative chronological order
3613 of expected time to complete (i.e., the current scheduled order). For efficiency, it is only necessary to
3614 calculate this value when an operation is performed that requests this attribute.

3615 4.3.14 job-message-from-operator (text(127))

3616 This attribute provides a message from an operator, system administrator or "intelligent" process to
3617 indicate to the end user the reasons for modification or other management action taken on a job.

3618 4.3.15 Job Size Attributes

3619 This sub-section defines job attributes that describe the size of the job. These attributes are not intended
3620 to be counters; they are intended to be useful routing and scheduling information if known. For these
3621 attributes, the Printer object may try to compute the value if it is not supplied in the create request. Even
3622 if the client does supply a value for these three attributes in the create request, the Printer object MAY
3623 choose to change the value if the Printer object is able to compute a value which is more accurate than
3624 the client supplied value. The Printer object may be able to determine the correct value for these
3625 attributes either right at job submission time or at any later point in time.

3626 4.3.15.1 job-k-octets (integer(0:MAX))

3627 This attribute specifies the total size of the document(s) in K octets, i.e., in units of 1024 octets requested
3628 to be processed in the job. The value MUST be rounded up, so that a job between 1 and 1024 octets
3629 MUST be indicated as being 1, 1025 to 2048 MUST be 2, etc.

3630 This value MUST NOT include the multiplicative factors contributed by the number of copies specified
3631 by the "copies" attribute, independent of whether the device can process multiple copies without making
3632 multiple passes over the job or document data and independent of whether the output is collated or not.
3633 Thus the value is independent of the implementation and indicates the size of the document(s) measured
3634 in K octets independent of the number of copies.

3635 This value MUST also not include the multiplicative factor due to a copies instruction embedded in the
3636 document data. If the document data actually includes replications of the document data, this value will
3637 include such replication. In other words, this value is always the size of the source document data, rather
3638 than a measure of the hardcopy output to be produced.

3639 ~~Note: This attribute and the following two attributes ("job-impressions" and "job-media-sheets") are not~~
3640 ~~intended to be counters; they are intended to be useful routing and scheduling information if known. For~~
3641 ~~these three attributes, the Printer object may try to compute the value if it is not supplied in the create~~
3642 ~~request. Even if the client does supply a value for these three attributes in the create request, the Printer~~
3643 ~~object MAY choose to change the value if the Printer object is able to compute a value which is more~~
3644 ~~accurate than the client supplied value. The Printer object may be able to determine the correct value for~~
3645 ~~these three attributes either right at job submission time or at any later point in time.~~

3646 4.3.15.2 job-impressions (integer(0:MAX))

3647 This attribute specifies the total size in number of impressions of the document(s) being submitted (see
3648 the definition of impression in section 12.2.5).

3649 As with "job-k-octets", this value MUST NOT include the multiplicative factors contributed by the
3650 number of copies specified by the "copies" attribute, independent of whether the device can process
3651 multiple copies without making multiple passes over the job or document data and independent of
3652 whether the output is collated or not. Thus the value is independent of the implementation and reflects
3653 the size of the document(s) measured in impressions independent of the number of copies.

3654 As with "job-k-octets", this value MUST also not include the multiplicative factor due to a copies
3655 instruction embedded in the document data. If the document data actually includes replications of the
3656 document data, this value will include such replication. In other words, this value is always the number
3657 of impressions in the source document data, rather than a measure of the number of impressions to be
3658 produced by the job.

3659 ~~See the Note in the "job-k-octets" attribute that also applies to this attribute.~~

3660 4.3.15.3 job-media-sheets (integer(0:MAX))

3661 This attribute specifies the total number of media sheets to be produced for this job.

3662 Unlike the "job-k-octets" and the "job-impressions" attributes, this value MUST include the
3663 multiplicative factors contributed by the number of copies specified by the "copies" attribute and a
3664 'number of copies' instruction embedded in the document data, if any. This difference allows the system
3665 administrator to control the lower and upper bounds of both (1) the size of the document(s) with "job-k-
3666 octets-supported" and "job-impressions-supported" and (2) the size of the job with "job-media-sheets-
3667 supported".

3668 ~~See the Note in the "job-k-octets" attribute that also applies to this attribute.~~

3669 4.3.16 Job Progress Attributes

3670 This sub-section defines job attributes that describe the progress of the job. These attributes are intended
3671 to be counters. That is, the value for a job that has not started processing MUST be 0. When the job's
3672 "job-state" is 'processing' or 'processing-stopped', this value is intended to contain the amount of the job
3673 that has been processed to the time at which the attributes are requested. When the job enters the
3674 'completed', 'canceled', or 'aborted' states, these values are the final values for the job.

3675 4.3.16.1 job-k-octets-processed (integer(0:MAX))

3676 This attribute specifies the total number of octets processed in K octets, i.e., in units of 1024 octets so
3677 far. The value MUST be rounded up, so that a job between 1 and 1024 octets inclusive MUST be
3678 indicated as being 1, 1025 to 2048 inclusive MUST be 2, etc.

3679 For implementations where multiple copies are produced by the interpreter with only a single pass over
3680 the data, the final value MUST be equal to the value of the "job-k-octets" attribute. For implementations
3681 where multiple copies are produced by the interpreter by processing the data for each copy, the final
3682 value MUST be a multiple of the value of the "job-k-octets" attribute.

3683 ~~Note: This attribute and the following two attributes ("job-impressions-completed" and "job-sheets-~~
3684 ~~completed") are intended to be counters. That is, the value for a job that has not started processing~~
3685 ~~MUST be 0. When the job's "job-state" is 'processing' or 'processing-stopped', this value is intended to~~
3686 ~~contain the amount of the job that has been processed to the time at which the attributes are requested.~~

3687 4.3.16.2 job-impressions-completed (integer(0:MAX))

3688 This job attribute specifies the number of impressions completed for the job so far. For printing devices,
3689 the impressions completed includes interpreting, marking, and stacking the output.

3690 ~~See the note in "job-k-octets-processed" which also applies to this attribute.~~

3691 4.3.16.3 job-media-sheets-completed (integer(0:MAX))

3692 This job attribute specifies the media-sheets completed marking and stacking for the entire job so far
3693 whether those sheets have been processed on one side or on both.

3694 ~~See the note in "job-k-octets-processed" which also applies to this attribute.~~

3695 4.3.17 attributes-charset (charset)

3696 This REQUIRED attribute is populated using the value in the client supplied "attributes-charset"
3697 attribute in the create request. It identifies the charset (coded character set and encoding method) used
3698 by any Job attributes with attribute syntax 'text' and 'name' that were supplied by the client in the create
3699 request. See Section 3.1.4 for a complete description of the "attributes-charset" operation attribute.

3700 This attribute does not indicate the charset in which the 'text' and 'name' values are stored internally in
3701 the Job object. The internal charset is implementation-defined. The IPP object MUST convert from
3702 whatever the internal charset is to that being requested in an operation as specified in Section 3.1.4.

3703 4.3.18 attributes-natural-language (naturalLanguage)

3704 This REQUIRED attribute is populated using the value in the client supplied "attributes-natural-
3705 language" attribute in the create request. It identifies the natural language used for any Job attributes
3706 with attribute syntax 'text' and 'name' that were supplied by the client in the create request. See Section
3707 3.1.4 for a complete description of the "attributes-natural-language" operation attribute. See Sections
3708 4.1.1.2 and 4.1.2.2 for how a Natural Language Override may be supplied explicitly for each 'text' and
3709 'name' attribute value that differs from the value identified by the "attributes-natural-language" attribute.

3710 4.4 Printer Description Attributes

3711 These attributes form the attribute group called "printer-description". The following table summarizes
3712 these attributes, their syntax, and whether or not they are REQUIRED for a Printer object to support. If
3713 they are not indicated as REQUIRED, they are OPTIONAL. The maximum size in octets for 'text' and
3714 'name' attributes is indicated in parentheses.

3715 Note: How these attributes are set by an Administrator is outside the scope of this IPP/1.1 document.

3716	+-----+-----+-----+
3717	Attribute Syntax REQUIRED?
3718	+-----+-----+-----+
3719	printer-uri-supported 1setOf uri REQUIRED
3720	+-----+-----+-----+
3721	uri-security-supported 1setOf type2 keyword REQUIRED
3722	+-----+-----+-----+
3723	<u>uri-authentication-supported</u> <u>1setOf type2 keyword</u> <u>REQUIRED</u>
3724	+-----+-----+-----+
3725	printer-name name (127) REQUIRED
3726	+-----+-----+-----+
3727	printer-location text (127)
3728	+-----+-----+-----+
3729	printer-info text (127)
3730	+-----+-----+-----+
3731	printer-more-info uri
3732	+-----+-----+-----+
3733	printer-driver-installer uri
3734	+-----+-----+-----+
3735	printer-make-and-model text (127)
3736	+-----+-----+-----+
3737	printer-more-info- uri
3738	manufacturer
3739	+-----+-----+-----+
3740	printer-state type1 enum REQUIRED
3741	+-----+-----+-----+
3742	printer-state-reasons 1setOf type2 keyword <u>REQUIRED</u>
3743	+-----+-----+-----+
3744	printer-state-message text (MAX)
3745	+-----+-----+-----+
3746	<u>ipp-versions-supported</u> <u>1setOf type2 keyword</u> <u>REQUIRED</u>
3747	+-----+-----+-----+
3748	<u>operations-supported</u> <u>1setOf type2 enum</u> <u>REQUIRED</u>
3749	+-----+-----+-----+
3750	<u>ipp-multiple-document-jobs-</u> <u>boolean</u>
3751	<u>supported</u>
3752	+-----+-----+-----+
3753	charset-configured charset REQUIRED
3754	+-----+-----+-----+
3755	charset-supported 1setOf charset REQUIRED
3756	+-----+-----+-----+
3757	natural-language-configured naturalLanguage REQUIRED
3758	+-----+-----+-----+
3759	generated-natural-language- 1setOf naturalLanguage REQUIRED
3760	supported
3761	+-----+-----+-----+
3762	document-format-default mimeType REQUIRED
3763	+-----+-----+-----+
3764	document-format-supported 1setOf mimeType REQUIRED
3765	+-----+-----+-----+

3766	printer-is-accepting-jobs	boolean	REQUIRED
3767	+-----+	+-----+	+-----+
3768	queued-job-count	integer (0:MAX)	<u>REQUIRED</u>
3769	<u>RECOMMENDED</u>		
3770	+-----+	+-----+	+-----+
3771	printer-message-from-	text (127)	
3772	operator		
3773	+-----+	+-----+	+-----+
3774	color-supported	boolean	
3775	+-----+	+-----+	+-----+
3776	reference-uri-schemes-	1setOf uriScheme	
3777	supported		
3778	+-----+	+-----+	+-----+
3779	pdl-override-supported	type2 keyword	REQUIRED
3780	+-----+	+-----+	+-----+
3781	printer-up-time	integer (1:MAX)	REQUIRED
3782	+-----+	+-----+	+-----+
3783	printer-current-time	dateTime	
3784	+-----+	+-----+	+-----+
3785	multiple-operation-time-out	integer (1:MAX)	
3786	+-----+	+-----+	+-----+
3787	compression-supported	1setOf type3 keyword	<u>REQUIRED</u>
3788	+-----+	+-----+	+-----+
3789	job-k-octets-supported	rangeOfInteger (0:MAX)	
3790	+-----+	+-----+	+-----+
3791	job-impressions-supported	rangeOfInteger (0:MAX)	
3792	+-----+	+-----+	+-----+
3793	job-media-sheets-supported	rangeOfInteger (0:MAX)	
3794	+-----+	+-----+	+-----+
3795	pages-per-minute	integer(0:MAX)	
3796	+-----+	+-----+	+-----+
3797	pages-per-minute-color	integer(0:MAX)	
3798	+-----+	+-----+	+-----+
3799			

3800 4.4.1 printer-uri-supported (1setOf uri)

3801 This REQUIRED Printer attribute contains at least one URI for the Printer object. It OPTIONALLY
3802 contains more than one URI for the Printer object. An administrator determines a Printer object's
3803 URI(s) and configures this attribute to contain those URIs by some means outside the scope of this
3804 IPP/1.1 document. The precise format of this URI is implementation dependent and depends on the
3805 protocol. See the next two sections for a description of the "uri-security-supported" and "uri-
3806 authentication-supported" attributes, both of which is-are the REQUIRED companion attributes to this
3807 "printer-uri-supported" attribute. See section 2.4 on Printer object identity and section 8.2 on security
3808 and URIs for more information.

3809 4.4.2 uri-authentication-supported (1setOf type2 keyword) **Issue 2**

3810 This REQUIRED Printer attribute MUST have the same cardinality (contain the same number of values)
3811 as the "printer-uri-supported" attribute. This attribute identifies the Client Authentication mechanism
3812 associated with each URI listed in the "printer-uri-supported" attribute. The Printer object uses the
3813 specified mechanism to identify the authenticated user (see section 8.3). The "i th" value in "uri-
3814 authentication-supported" corresponds to the "i th" value in "printer-uri-supported" and it describes the
3815 authentication mechanisms used by the Printer when associated with accessed via that URI. See [IPP-
3816 PRO] for more details on Client Authentication.

3817 The following standard keyword values are defined:

3818 'none': There is no authentication mechanism associated with the URI. The Printer object assumes
3819 that the authenticated user is "anonymous".
3820 'requesting-user-name': When a client performs an operation whose target is the associated URI, tThe
3821 Printer object assumes that the authenticated user is specified by the "requesting-user-name"
3822 Operation attribute (see section 8.3). If this the "requesting-user-name" attribute is absent in a
3823 request, the Printer object assumes that the authenticated user is "anonymous".
3824 'basic': When a client performs an operation whose target is the associated URI, the Printer object
3825 challenges the client with HTTP basic authentication. The Printer object assumes that the
3826 authenticated user is the name received via the basic authentication mechanism. This
3827 authentication mechanism SHOULD be used with a secure channel, that is, the corresponding
3828 value of "uri-security-supported" SHOULD NOT be 'none'.
3829 'digest': When a client performs an operation whose target is the associated URI, the Printer object
3830 challenges the client with HTTP digest authentication. The Printer object assumes that the
3831 authenticated user is the name received via the digest authentication mechanism.
3832 'certificate': When a client performs an operation whose target is the associated URI, the Printer
3833 object expects the client to provide a certificate. The Printer object assumes that the authenticated
3834 user is the textual name contained within the certificate.

3835 4.4.3 uri-security-supported (1setOf type2 keyword)

3836 This REQUIRED Printer attribute MUST have the same cardinality (contain the same number of values)
3837 as the "printer-uri-supported" attribute. This attribute identifies the security mechanisms used for each
3838 URI listed in the "printer-uri-supported" attribute. The "i th" value in "uri-security-supported"
3839 corresponds to the "i th" value in "printer-uri-supported" and it describes the security mechanisms used
3840 for accessing the Printer object via that URI. See [IPP-PRO] for more details on security mechanisms.

3841 The following standard keyword values are defined:

3842 'none': There are no secure communication channel protocols in use for the given URI.
3843 'ssl3': SSL3 [SSL] is the secure communications channel protocol in use for the given URI. For use
3844 in IPP/1.0.
3845 'tls': TLS [RFC2246] is the secure communications channel protocol in use for the given URI. For
3846 use in IPP/1.1.
3847

3848 This attribute is orthogonal to the specification of a Client Authentication mechanism. Specifically,
 3849 'none' does not exclude Client Authentication. See section 4.4.2. Issue 21

3850 Consider the following example. For a single Printer object, an administrator configures the "printer-uri-
 3851 supported", "uri-authentication-supported" and "uri-security-supported" attributes as follows:

3852 "printer-uri-supported": 'xxx://acme.com/open-use-printer', 'xxx://acme.com/restricted-use-printer',
 3853 'xxx://acme.com/private-printer'
 3854 "uri-authentication-supported": 'none', 'digest', 'basic'
 3855 "uri-security-supported": 'none', 'none', 'tls'
 3856

3857 Note: 'xxx' is not a valid scheme. See the IPP/1.1 "Transport and Encoding" specification [IPP-PRO]
 3858 for the actual URI schemes to be used in object target attributes.

3859 In this case, one Printer object has three URIs.

- 3860 - For the first URI, 'xxx://acme.com/open-use-printer', the value 'none' in "uri-security-supported"
 3861 indicates that there is no secure channel protocol configured to run under HTTP. The value of
 3862 'none' in "uri-authentication-supported" indicates that all users are "anonymous". There will be
 3863 no challenge and the Printer will ignore "requesting-user-name". The name implies that there is
 3864 no Basic or Digest authentication being used, but it is up to the client to determine that while
 3865 using HTTP underneath the IPP application protocol.
- 3866 - For the second URI, 'xxx://acme.com/restricted-use-printer', the value 'none' in "uri-security-
 3867 supported" indicates that there is no secure channel protocol configured to run under HTTP. The
 3868 value of 'digest' in "uri-authentication-supported" indicates that the Printer will issue a challenge
 3869 and that the Printer will use the name supplied by the digest mechanism to determine the
 3870 authenticated user (see section 8.3). In this case, although the name does imply that there is some
 3871 sort of Basic or Digest authentication being used within HTTP, it is up to the client to determine
 3872 that while using HTTP and by processing any '401 Unauthorized' HTTP error messages.
- 3873 - For the third URI, 'xxx://acme.com/private-printer', the value 'tls' in "uri-security-supported"
 3874 indicates that TLS is being used to secure the channel. The client SHOULD be prepared to use
 3875 TLS framing to negotiate an acceptable ciphersuite to use while communicating with the Printer
 3876 object. In this case, the name implies the use of a secure communications channel, but the fact is
 3877 made explicit by the presence of the 'tls' value in "uri-security-supported". The client does not
 3878 need to resort to understanding which security it must use by following naming conventions or by
 3879 parsing the URI to determine which security mechanisms are implied. The value of 'basic' in "uri-
 3880 authentication-supported" indicates that the Printer will issue a challenge and that the Printer will
 3881 use the name supplied by the digest mechanism to determine the authenticated user (see section
 3882 8.3). Because this challenge occurs in a tls session, the channel is secure.

3884 It is expected that many IPP Printer objects will be configured to support only one channel (either
 3885 configured to use TLS access or not) and only one authentication mechanism. , and will Such Printer
 3886 objects therefore only ever have one URI listed in the "printer-uri-supported" attribute. No matter the
 3887 configuration of the Printer object (whether it has only one URI or more than one URI), a client MUST
 3888 supply only one URI in the target "printer-uri" operation attribute.

3889 4.4.4 printer-name (name(127))

3890 This REQUIRED Printer attribute contains the name of the Printer object. It is a name that is more end-
3891 user friendly than a URI. An administrator determines a printer's name and sets this attribute to that
3892 name. This name may be the last part of the printer's URI or it may be unrelated. In non-US-English
3893 locales, a name may contain characters that are not allowed in a URI.

3894 4.4.5 printer-location (text(127))

3895 This Printer attribute identifies the location of the device. This could include things like: "in Room
3896 123A, second floor of building XYZ".

3897 4.4.6 printer-info (text(127))

3898 This Printer attribute identifies the descriptive information about this Printer object. This could include
3899 things like: "This printer can be used for printing color transparencies for HR presentations", or "Out of
3900 courtesy for others, please print only small (1-5 page) jobs at this printer", or even "This printer is going
3901 away on July 1, 1997, please find a new printer".

3902 4.4.7 printer-more-info (uri)

3903 This Printer attribute contains a URI used to obtain more information about this specific Printer object.
3904 For example, this could be an HTTP type URI referencing an HTML page accessible to a Web Browser.
3905 The information obtained from this URI is intended for end user consumption. Features outside the
3906 scope of IPP can be accessed from this URI. The information is intended to be specific to this printer
3907 instance and site specific services (e.g. job pricing, services offered, end user assistance). The device
3908 manufacturer may initially populate this attribute.

3909 4.4.8 printer-driver-installer (uri)

3910 This Printer attribute contains a URI to use to locate the driver installer for this Printer object. This
3911 attribute is intended for consumption by automata. The mechanics of print driver installation is outside
3912 the scope of this IPP/1.1 document. The device manufacturer may initially populate this attribute.

3913 4.4.9 printer-make-and-model (text(127))

3914 This Printer attribute identifies the make and model of the device. The device manufacturer may
3915 initially populate this attribute.

3916 4.4.10 printer-more-info-manufacturer (uri)

3917 This Printer attribute contains a URI used to obtain more information about this type of device. The
3918 information obtained from this URI is intended for end user consumption. Features outside the scope of
3919 IPP can be accessed from this URI (e.g., latest firmware, upgrades, print drivers, optional features
3920 available, details on color support). The information is intended to be germane to this printer without

3921 regard to site specific modifications or services. The device manufacturer may initially populate this
3922 attribute.

3923 4.4.11 printer-state (type1 enum)

3924 This REQUIRED Printer attribute identifies the current state of the device. The "printer-state reasons"
3925 attribute augments the "printer-state" attribute to give more detailed information about the Printer in the
3926 given printer state.

3927 A Printer object need only update this attribute before responding to an operation which requests the
3928 attribute; the Printer object NEED NOT update this attribute continually, since asynchronous event
3929 notification is not part of IPP/1.1. A Printer NEED NOT implement all values if they are not applicable
3930 to a given implementation.

3931 The following standard enum values are defined:

3932 Value	Symbolic Name and Description
------------	-------------------------------

3933 3934 3935 3936 3937 3938	'3' 'idle': If a Printer receives a job (whose required resources are ready) while in this state, such a job MUST transit into the 'processing' state immediately. If the "printer-state-reasons" attribute contains any reasons, they MUST be reasons that would not prevent a job from transiting into the 'processing' state immediately, e.g., 'toner-low'.
--	---

If a Printer can interpret one or more jobs while marking a job, then it is idle if it is available to interpret jobs even while marking a job. Issue 31

3942
3943 Note: If a Printer controls more than one output device, the above definition implies that a Printer is 'idle' if at least one output device is idle, i.e., the IPP Printer is available to immediately start processing a job if a client submitted it.

3946 3947 3948 3949 3950 3951 3952	'4' 'processing': If a Printer receives a job (whose required resources are ready) while in this state, such a job MUST transit into the 'pending' state immediately. Such a job MUST transit into the 'processing' state only after jobs ahead of it complete. If the "printer-state-reasons" attribute contains any reasons, they MUST be reasons that do not prevent the current job from printing, e.g. 'toner-low'.
--	--

If a Printer can interpret one or more jobs while marking a job and receives a job (whose required resources are ready) while in this state, such a received job MAY transit into the 'processing' state along with the job that is being marked, if any. Issue 31

3956
3957
3958 Note: If a Printer controls more than one output device, the above definition implies that a Printer is 'processing' if at least one output device is processing, and none is idle.

3961

3962 '5' 'stopped': If a Printer receives a job (whose required resources are ready) while in this
 3963 state, such a job MUST transit into the 'pending' state immediately. Such a job
 3964 MUST transit into the 'processing' state only after some human fixes the problem
 3965 that stopped the printer and after jobs ahead of it complete processing. ~~If~~
 3966 ~~supported.~~ Issue 30 The "printer-state-reasons" attribute MUST contain at least
 3967 one reason, e.g. 'media-jam', which prevents it from either processing the current
 3968 job or transitioning a 'pending' job to the 'processing' state.

If a Printer can interpret one or more jobs while marking a job and receives a job
 (whose required resources are ready) while in this state, such a submitted job
 MAY transit into the 'processing' state in order to be interpreted even while the
 Printer is in the 'stopped' state. However, before such a job can be completed, a
 human needs to fix the problem. Issue 31

~~Note:~~ ~~i~~If a Printer controls more than one output device, the above definition
 implies that a Printer is 'stopped' only if all output devices are stopped.

~~Also, Note:~~ it is tempting to define 'stopped' as when a sufficient number of output
 devices are stopped and leave it to an implementation to define the sufficient
 number. But such a rule complicates the definition of 'stopped' and 'processing'.
 For example, with this alternate definition of 'stopped', a job can move from
 'pending' to 'processing' without human intervention, even though the Printer is
 stopped.

3986 4.4.12 printer-state-reasons (1setOf type2 keyword)

3987 This **REQUIRED** Printer attribute supplies additional detail about the device's state. Issue 30

3988 Each keyword value MAY have a suffix to indicate its level of severity. The three levels are: report
 3989 (least severe), warning, and error (most severe).

- 3990 - '-report': This suffix indicates that the reason is a "report". An implementation may choose to omit
 3991 some or all reports. Some reports specify finer granularity about the printer state; others serve as
 3992 a precursor to a warning. A report MUST contain nothing that could affect the printed output.
- 3993 - '-warning': This suffix indicates that the reason is a "warning". An implementation may choose to
 3994 omit some or all warnings. Warnings serve as a precursor to an error. A warning MUST contain
 3995 nothing that prevents a job from completing, though in some cases the output may be of lower
 3996 quality.
- 3997 - '-error': This suffix indicates that the reason is an "error". An implementation MUST include all
 3998 errors. If this attribute contains one or more errors, printer MUST be in the stopped state.

4000 If the implementation does not add any one of the three suffixes, all parties MUST assume that the
 4001 reason is an "error".

4002 If a Printer object controls more than one output device, each value of this attribute MAY apply to one or
4003 more of the output devices. An error on one output device that does not stop the Printer object as a
4004 whole MAY appear as a warning in the Printer's "printer-state-reasons" attribute. If the "printer-state"
4005 for such a Printer has a value of 'stopped', then there MUST be an error reason among the values in the
4006 "printer-state-reasons" attribute.

4007 The following standard keyword values are defined:

4008 'other': The device has detected an error other than one listed in this document.

4009 'none': There are not reasons. This state reason is semantically equivalent to "printer-state-reasons"
4010 without any value and MUST be used, since the 1setOf attribute syntax requires at least one
4011 value.

4012 'media-needed': A tray has run out of media.

4013 'media-jam': The device has a media jam.

4014 'moving-to-paused': Someone has paused the Printer object using the Pause-Printer operation (see
4015 section 3.2.7) or other means, but the device(s) are taking an appreciable time to stop. Later,
4016 when all output has stopped, the "printer-state" becomes 'stopped', and the 'paused' value replaces
4017 the 'moving-to-paused' value in the "printer-state-reasons" attribute.

4018 'paused': Someone has paused the Printer object using the Pause-Printer operation (see section 3.2.7)
4019 or other means and the Printer object's "printer-state" is 'stopped'. In this state, a Printer MUST
4020 NOT produce printed output, but it MUST perform other operations requested by a client. If a
4021 Printer had been printing a job when the Printer was paused, the Printer MUST resume printing
4022 that job when the Printer is no longer paused and leave no evidence in the printed output of such
4023 a pause.

4024 'shutdown': Someone has removed a Printer object from service, and the device may be powered
4025 down or physically removed. In this state, a Printer object MUST NOT produce printed output,
4026 and unless the Printer object is realized by a print server that is still active, the Printer object
4027 MUST perform no other operations requested by a client, including returning this value. If a
4028 Printer object had been printing a job when it was shutdown, the Printer NEED NOT resume
4029 printing that job when the Printer is no longer shutdown. If the Printer resumes printing such a
4030 job, it may leave evidence in the printed output of such a shutdown, e.g. the part printed before
4031 the shutdown may be printed a second time after the shutdown.

4032 'connecting-to-device': The Printer object has scheduled a job on the output device and is in the
4033 process of connecting to a shared network output device (and might not be able to actually start
4034 printing the job for an arbitrarily long time depending on the usage of the output device by other
4035 servers on the network).

4036 'timed-out': The server was able to connect to the output device (or is always connected), but was
4037 unable to get a response from the output device.

4038 'stopping': The Printer object is in the process of stopping the device and will be stopped in a while.
4039 When the device is stopped, the Printer object will change the Printer object's state to 'stopped'.
4040 The 'stopping-warning' reason is never an error, even for a Printer with a single output device.
4041 When an output-device ceases accepting jobs, the Printer will have this reason while the output
4042 device completes printing.

4043 'stopped-partly': When a Printer object controls more than one output device, this reason indicates
4044 that one or more output devices are stopped. If the reason is a report, fewer than half of the

4045 output devices are stopped. If the reason is a warning, fewer than all of the output devices are
4046 stopped.

4047 'toner-low': The device is low on toner.

4048 'toner-empty': The device is out of toner.

4049 'spool-area-full': The limit of persistent storage allocated for spooling has been reached.

4050 'cover-open': One or more covers on the device are open.

4051 'interlock-open': One or more interlock devices on the printer are unlocked.

4052 'door-open': One or more doors on the device are open.

4053 'input-tray-missing': One or more input trays are not in the device.

4054 'media-low': At least one input tray is low on media.

4055 'media-empty': At least one input tray is empty.

4056 'output-tray-missing': One or more output trays are not in the device

4057 'output-area-almost-full': One or more output area is almost full (e.g. tray, stacker, collator).

4058 'output-area-full': One or more output area is full. (e.g. tray, stacker, collator)

4059 'marker-supply-low': The device is low on at least one marker supply. (e.g. toner, ink, ribbon)

4060 'marker-supply-empty': The device is out of at least one marker supply. (e.g. toner, ink, ribbon)

4061 'marker-waste-almost-full': The device marker supply waste receptacle is almost full.

4062 'marker-waste-full': The device marker supply waste receptacle is full.

4063 'fuser-over-temp': The fuser temperature is above normal.

4064 'fuser-under-temp': The fuser temperature is below normal.

4065 'opc-near-eol': The optical photo conductor is near end of life.

4066 'opc-life-over': The optical photo conductor is no longer functioning.

4067 'developer-low': The device is low on developer.

4068 'developer-empty': The device is out of developer.

4069 'interpreter-resource-unavailable': An interpreter resource is unavailable (i.e. font, form)

4070

4071 4.4.13 printer-state-message (text(MAX))

4072 This Printer attribute specifies the additional information about the printer state and printer state reasons
4073 in human readable text. If the Printer object supports this attribute, the Printer object MUST be able to
4074 generate this message in any of the natural languages identified by the Printer's "generated-natural-
4075 language-supported" attribute (see the "attributes-natural-language" operation attribute specified in
4076 Section 3.1.4.1).

4077 ~~4.5.14~~4.4.14 ipp-versions-supported (1setOf type2 keyword) Issue 36

4078 This REQUIRED attribute identifies the IPP protocol versions that this Printer supports, including minor
4079 versions, i.e., the values of the "version-number" parameter that it will accept in requests and return in
4080 responses. If an IPP Printer receives a request with the "version-number" parameter set to a (two-octet
4081 binary) value that does not correspond to one of the values of this (US-ASCII) keyword, it MUST reject
4082 the request and return the 'server-error-version-not-supported' status code. See Section 3.1.8.

4083 The following standard keyword values are defined:

4084 '1.0': Version 1.0 as specified in RFC 2566 [RFC2566] and RFC 2565 [RFC2565] including any
4085 extensions registered according to Section 6 and any extension defined in this version or any

4086 future version of this document following the rules when the "version-number" parameter is '1.0',
 4087 if any. For an example of such a '1.0' rule, see section 4.3.12.
 4088 '1.1': Version 1.1 as specified in this document and [IPP-PRO] including any extensions registered
 4089 according to Section 6 or defined in any future version of this document following the rules when
 4090 the "version-number" parameter is '1.1', if any.

4091 4.4.15 operations-supported (1setOf type2 enum)

4092 This REQUIRED Printer attribute specifies the set of supported operations for this Printer object and
 4093 contained Job objects.

4094 Note: This attribute is encoded as any other enum attribute syntax according to [IPP-PRO] as 32-bits.
 4095 However, all 32-bit enum values for this attribute MUST NOT exceed 0x00008FFF, since these same
 4096 values are also passed in two octets in the "operation-id" parameter (see section 3.1.1) in each Protocol
 4097 request with the two high order octets omitted in order to indicate the operation being performed [IPP-
 4098 PRO].

4099 The following standard enum and "operation-id" (see section 3.1.2) values are defined:

4100	Value	Operation Name
4101	-----	-----
4102		
4103	0x0000	reserved, not used
4104	0x0001	reserved, not used
4105	0x0002	Print-Job
4106	0x0003	Print-URI
4107	0x0004	Validate-Job
4108	0x0005	Create-Job
4109	0x0006	Send-Document
4110	0x0007	Send-URI
4111	0x0008	Cancel-Job
4112	0x0009	Get-Job-Attributes
4113	0x000A	Get-Jobs
4114	0x000B	Get-Printer-Attributes
4115	0x000C	Hold-Job
4116	0x000D	Release-Job
4117	0x000E	Restart-Job
4118	0x000F	reserved for a future operation
4119	0x0010	Pause-Printer
4120	0x0011	Resume-Printer
4121	0x0012	Purge-Jobs
4122	0x00013-0x3FFF	reserved for future operations
4123	0x4000-0x8FFF	reserved for private extensions
4124		

4125 This reserved block for private extensions allows for ~~certain~~ vendors to implement private extensions
4126 that are guaranteed to not conflict with future registered extensions. However, there is no guarantee that
4127 two or more private extensions will not conflict.

4128 ~~4.5.16~~4.4.16 multiple-document-jobs-supported (boolean) **Issue 34**

4129 This Printer attribute indicates whether or not the Printer supports more than one document per job, i.e.,
4130 more than one Send-Document or Send-Data operation with document data. If the Printer supports the
4131 Create-Job and Send-Document operations (see section 3.2.4 and 3.3.1), it MUST support this attribute.

4132 4.4.17 charset-configured (charset)

4133 This REQUIRED Printer attribute identifies the charset that the Printer object has been configured to
4134 represent 'text' and 'name' Printer attributes that are set by the operator, system administrator, or
4135 manufacturer, i.e., for "printer-name" (name), "printer-location" (text), "printer-info" (text), and "printer-
4136 make-and-model" (text). Therefore, the value of the Printer object's "charset-configured" attribute
4137 MUST also be among the values of the Printer object's "charset-supported" attribute.

4138 4.4.18 charset-supported (1setOf charset)

4139 This REQUIRED Printer attribute identifies the set of charsets that the Printer and contained Job objects
4140 support in attributes with attribute syntax 'text' and 'name'. At least the value 'utf-8' MUST be present,
4141 since IPP objects MUST support the UTF-8 [RFC2279] charset. If a Printer object supports a charset, it
4142 means that for all attributes of syntax 'text' and 'name' the IPP object MUST (1) accept the charset in
4143 requests and return the charset in responses as needed.

4144 If more charsets than UTF-8 are supported, the IPP object MUST perform charset conversion between
4145 the charsets as described in Section 3.1.4.23.2.1.2.

4146 4.4.19 natural-language-configured (naturalLanguage)

4147 This REQUIRED Printer attribute identifies the natural language that the Printer object has been
4148 configured to represent 'text' and 'name' Printer attributes that are set by the operator, system
4149 administrator, or manufacturer, i.e., for "printer-name" (name), "printer-location" (text), "printer-info"
4150 (text), and "printer-make-and-model" (text). When returning these Printer attributes, the Printer object
4151 MAY return them in the configured natural language specified by this attribute, instead of the natural
4152 language requested by the client in the "attributes-natural-language" operation attribute. See Section
4153 3.1.4.1 for the specification of the OPTIONAL multiple natural language support. Therefore, the value
4154 of the Printer object's "natural-language-configured" attribute MUST also be among the values of the
4155 Printer object's "generated-natural-language-supported" attribute.

4156 4.4.20 generated-natural-language-supported (1setOf naturalLanguage)

4157 This REQUIRED Printer attribute identifies the natural language(s) that the Printer object and contained
4158 Job objects support in attributes with attribute syntax 'text' and 'name'. The natural language(s)

4159 supported depends on implementation and/or configuration. Unlike charsets, IPP objects MUST accept
4160 requests with any natural language or any Natural Language Override whether the natural language is
4161 supported or not.

4162 If a Printer object supports a natural language, it means that for any of the attributes for which the Printer
4163 or Job object generates messages, i.e., for the "job-state-message" and "printer-state-message" attributes
4164 and Operation Messages (see Section 3.1.5) in operation responses, the Printer and Job objects MUST be
4165 able to generate messages in any of the Printer's supported natural languages. See section 3.1.4 for the
4166 specification of 'text' and 'name' attributes in operation requests and responses.

4167 Note: A Printer object that supports multiple natural languages, often has separate catalogs of messages,
4168 one for each natural language supported.

4169 4.4.21 document-format-default (mimeMediaType)

4170 This REQUIRED Printer attribute identifies the document format that the Printer object has been
4171 configured to assume if the client does not supply a "document-format" operation attribute in any of the
4172 operation requests that supply document data. The standard values for this attribute are Internet Media
4173 types (sometimes called MIME types). For further details see the description of the 'mimeMediaType'
4174 attribute syntax in Section 4.1.9.

4175 4.4.22 document-format-supported (1setOf mimeMediaType)

4176 This REQUIRED Printer attribute identifies the set of document formats that the Printer object and
4177 contained Job objects can support. For further details see the description of the 'mimeMediaType'
4178 attribute syntax in Section 4.1.9.

4179 4.4.23 printer-is-accepting-jobs (boolean)

4180 This REQUIRED Printer attribute indicates whether the printer is currently able to accept jobs, i.e., is
4181 accepting Print-Job, Print-URI, and Create-Job requests. If the value is 'true', the printer is accepting
4182 jobs. If the value is 'false', the Printer object is currently rejecting any jobs submitted to it. In this case,
4183 the Printer object returns the 'server-error-not-accepting-jobs' status code.

4184 Note: This value is independent of the "printer-state" and "printer-state-reasons" attributes because its
4185 value does not affect the current job; rather it affects future jobs. This attribute may cause the Printer to
4186 reject jobs when the "printer-state" is 'idle' or it may cause the Printer object to accept jobs when the
4187 "printer-state" is 'stopped'.

4188 4.4.24 queued-job-count (integer(0:MAX))

4189 This ~~REQUIRED~~ ~~RECOMMENDED~~ Printer attribute contains a count of the number of jobs that are
4190 either 'pending', 'processing', 'pending-held', or 'processing-stopped' and is set by the Printer object. **Issue**
4191 **29**

4192 4.4.25 printer-message-from-operator (text(127))

4193 This Printer attribute provides a message from an operator, system administrator or "intelligent" process
4194 to indicate to the end user information or status of the printer, such as why it is unavailable or when it is
4195 expected to be available.

4196 4.4.26 color-supported (boolean)

4197 This Printer attribute identifies whether the device is capable of any type of color printing at all,
4198 including highlight color. All document instructions having to do with color are embedded within the
4199 document PDL (none are external IPP attributes in IPP/1.1).

4200 Note: end-users are able to determine the nature and details of the color support by querying the
4201 "printer-more-info-manufacturer" Printer attribute.

4202 4.4.27 reference-uri-schemes-supported (1setOf uriScheme)

4203 This Printer attribute specifies which URI schemes are supported for use in the "document-uri" operation
4204 attribute of the Print-URI or Send-URI operation. If a Printer object supports these optional operations,
4205 it MUST support the "reference-uri-schemes-supported" Printer attribute with at least the following
4206 schemed URI value:

4207 'ftp': The Printer object will use an FTP 'get' operation as defined in RFC 2228 [RFC2228] using
4208 FTP URLs as defined by [RFC2396] and [RFC2316].

4209
4210 The Printer object MAY OPTIONALLY support other URI schemes (see section 4.1.6).

4211 4.4.28 pdl-override-supported (type2 keyword)

4212 This REQUIRED Printer attribute expresses the ability for a particular Printer implementation to either
4213 attempt to override document data instructions with IPP attributes or not.

4214 This attribute takes on the following values:

- 4215 - 'attempted': This value indicates that the Printer object attempts to make the IPP attribute values
4216 take precedence over embedded instructions in the document data, however there is no guarantee.
- 4217 - 'not-attempted': This value indicates that the Printer object makes no attempt to make the IPP
4218 attribute values take precedence over embedded instructions in the document data.

4219
4220 Section 15 contains a full description of how this attribute interacts with and affects other IPP attributes,
4221 especially the "ipp-attribute-fidelity" attribute.

4222 4.4.29 printer-up-time (integer(1:MAX))

4223 This REQUIRED Printer attribute indicates the amount of time (in seconds) that this [Printer](#) instance of
4224 [this Printer implementation](#) has been up and running. [The value is a monotonically increasing value](#)

4225 ~~starting from 1 when the Printer object is started-up (initialized, booted, etc.). This value or the value of~~
4226 ~~"printer-current-time" is used to populate the Event Time Job Description attributes "time-at-creation",~~
4227 ~~"time-at-processing", and "time-at-completed", depending on implementation (see Section 4.3.12).~~
4228 ~~These time values are all measured in seconds and all have meaning only relative to this attribute,~~
4229 ~~"printer-up-time". The value is a monotonically increasing value starting from 1 when the Printer object~~
4230 ~~is started-up (initialized, booted, etc.).~~

4231 ~~If the Printer object software ceases running goes down at some value 'n', and restarts comes back up~~
4232 ~~without knowing the last value for "printer-up-time", the implementation MAY MUST reset this value~~
4233 ~~to 1. However, if the device or devices that the Printer object is representing are restarted or power~~
4234 ~~cycled, the Printer object MAY continue counting this value or MAY reset this value to 1 depending on~~
4235 ~~implementation. If this value is reset and the implementation has persistent jobs and the Event Time Job~~
4236 ~~Description Attributes are represented using the 'integer' form (instead of the 'dateTime' form), they~~
4237 ~~MUST be reset according to Section 4.3.12. Issue 17~~

4238 ~~1. Know how long it has been down, and resume at some value greater than 'n', or~~
4239 ~~2. Restart from 1.~~

4241 ~~In the first case, the Printer SHOULD not tweak any existing related Job attributes ("time-at-creation",~~
4242 ~~"time-at-processing", and "time-at-completed"). In the second case, the Printer object SHOULD reset~~
4243 ~~those attributes to 0. If a client queries a time-related Job attribute and finds the value to be 0, the client~~
4244 ~~MUST assume that the Job was submitted in some life other than the Printer's current life.~~

4245 4.4.30 printer-current-time (dateTime)

4246 This Printer attribute indicates the current ~~absolute~~ wall-clock time. ~~This value or the value of "printer-~~
4247 ~~uptime-time" is used to populate the Job attributes "time-at-creation", "time-at-processing", and "time-at-~~
4248 ~~completed", depending on implementation (see Section 4.3.12). If an implementation supports this~~
4249 ~~attribute, then a client could calculate the absolute wall clock time each Job's "time-at-creation", "time-~~
4250 ~~at-processing", and "time-at-completed" attributes by using both "printer-up-time" and this attribute,~~
4251 ~~"printer-current-time". If an implementation does not support this attribute, a client can only calculate~~
4252 ~~the relative time of certain events based on the REQUIRED "printer-up-time" attribute.~~

4253 ~~The date and time is obtained on a "best efforts basis" and does not have to be that precise in order to~~
4254 ~~work in practice. A Printer implementation sets the value of this attribute by obtaining the date and time~~
4255 ~~via some implementation-dependent means, such as getting the value from a network time server,~~
4256 ~~initialization at time of manufacture, or setting by an administrator. See [IPP-IIG] for examples. If an~~
4257 ~~implementation supports this attribute and the implementation knows that it has not yet been set to a~~
4258 ~~correct value, then the implementation MUST return the value of this attribute using the out-of-band 'no-~~
4259 ~~value' meaning not configured. See the beginning of section 4.1. Issue 17~~

4260 ~~The time zone of this attribute NEED NOT be the time zone used by people located near the Printer~~
4261 ~~object or device. The client MUST NOT expect that the time zone of any received 'dateTime' value to~~
4262 ~~be in the time zone of the client or in the time zone of the people located near the printer. Issue 17~~

4263 The client SHOULD display any dateTime attributes to the user in client local time by converting the
4264 'dateTime' value returned by the server to the time zone of the client, rather than using the time zone
4265 returned by the Printer in attributes that use the 'dateTime' attribute syntax. Issue 17

4266 4.4.31 multiple-operation-time-out (integer(1:MAX))

4267 This Printer attribute identifies the minimum time (in seconds) that the Printer object waits for
4268 additional Send-Document or Send-URI operations to follow a still-open multi-document Job object
4269 before taking any recovery actions, such as the ones indicated in section 3.3.1. If the Printer object
4270 supports the Create-Job and Send-Document operations (see section 3.2.4 and 3.3.1), it MUST support
4271 this attribute.

4272 It is RECOMMENDED that vendors supply a value for this attribute that is between 60 and 240
4273 seconds. An implementation MAY allow a system administrator to set this attribute (by means outside
4274 this IPP/1.1 document). If so, the system administrator MAY be able to set values outside this range.

4275 4.4.32 compression-supported (1setOf type3 keyword)

4276 This **REQUIRED** Printer attribute identifies the set of supported compression algorithms for document
4277 data. Compression only applies to the document data; compression does not apply to the encoding of the
4278 IPP operation itself. The supported values are used to validate the client supplied "compression"
4279 operation attributes in Print-Job, Send-Document, and Send-URI requests. Issue 28

4280 Standard values are :

- 4281 'none': no compression is used.
- 4282 'deflate': ZIP public domain inflate/deflate) compression technology
- 4283 'gzip' GNU zip compression technology described in RFC 1952 [RFC1952].
- 4284 'compress': UNIX compression technology

4285

4286 4.4.33 job-k-octets-supported (rangeOfInteger(0:MAX))

4287 This Printer attribute specifies the upper and lower bounds of total sizes of jobs in K octets, i.e., in units
4288 of 1024 octets. The supported values are used to validate the client supplied "job-k-octets" operation
4289 attributes in create requests. The corresponding job description attribute "job-k-octets" is defined in
4290 section 4.3.15.1.

4291 4.4.34 job-impressions-supported (rangeOfInteger(0:MAX))

4292 This Printer attribute specifies the upper and lower bounds for the number of impressions per job. The
4293 supported values are used to validate the client supplied "job-impressions" operation attributes in create
4294 requests. The corresponding job description attribute "job-impressions" is defined in section 4.3.15.2.

4295 4.4.35 job-media-sheets-supported (rangeOfInteger(0:MAX))

4296 This Printer attribute specifies the upper and lower bounds for the number of media sheets per job. The
4297 supported values are used to validate the client supplied "job-media-sheets" operation attributes in create
4298 requests. The corresponding Job attribute "job-media-sheets" is defined in section 4.3.15.3.

4299 4.4.36 pages-per-minute (integer(0:MAX))

4300 This Printer attributes specifies the nominal number of pages per minute to the nearest whole number
4301 which may be generated by this printer (e.g., simplex, black-and-white). This attribute is informative,
4302 not a service guarantee. Generally, it is the value used in the marketing literature to describe the device.

4303 A value of 0 indicates a device that takes more than two minutes to process a page.

4304 4.4.37 pages-per-minute-color (integer(0:MAX))

4305 This Printer attributes specifies the nominal number of pages per minute to the nearest whole number
4306 which may be generated by this printer when printing color (e.g., simplex, color). For purposes of this
4307 attribute, "color" means the same as for the "color-supported" attribute, namely, the device is capable of
4308 any type of color printing at all, including highlight color. This attribute is informative, not a service
4309 guarantee. Generally, it is the value used in the marketing literature to describe the color capabilities of
4310 this device.

4311 A value of 0 indicates a device that takes more than two minutes to process a page.

4312 Note: If a color device has several color modes, it MAY use the pages-per-minute value for this
4313 attribute that corresponds to the mode that produces the highest number.

4314 Black and white only printers MUST NOT support this attribute. If this attribute is present, then the
4315 "color-supported" Printer description attribute MUST be present and have a 'true' value.

4316 Note: The values of these two attributes returned by the Get-Printer-Attributes operation MAY be
4317 affected by the "document-format" attribute supplied by the client in the Get-Printer-Attributes request.
4318 In other words, the implementation MAY have different speeds depending on the document format
4319 being processed. See section 3.2.5.1 Get-Printer-Attributes.

4320 5. Conformance

4321 This section describes conformance issues and requirements. This document introduces model entities
4322 such as objects, operations, attributes, attribute syntaxes, and attribute values. These conformance
4323 sections describe the conformance requirements which apply to these model entities.

4324 5.1 Client Conformance Requirements

4325 This section describes the conformance requirements for a client (see section 2.1), whether it be:

4326 1. contained within software controlled by an end user, e.g. activated by the "Print" menu item in an
4327 application or

4328 2. a component of a print server that communicates (using IPP operations) with either an output
4329 device or another "downstream" print server. Issue 4 and Issue 5

4330 A conforming client MUST support all REQUIRED operations as defined in this document. For each
4331 attribute included in an operation request, a conforming client MUST supply a value whose type and
4332 value syntax conforms to the requirements of the Model document as specified in Sections 3 and 3.3.5.
4333 A conforming client MAY supply any registered extensions and/or private extensions in an operation
4334 request, as long as they meet the requirements in Section 6.

4335 Otherwise, there are no conformance requirements placed on the user interfaces provided by IPP clients
4336 or their applications. For example, one application might not allow an end user to submit multiple
4337 documents per job, while another does. One application might first query a Printer object in order to
4338 supply a graphical user interface (GUI) dialogue box with supported and default values whereas a
4339 different implementation might not.

4340 When sending a request, an IPP client NEED NOT supply any attributes that are indicated as
4341 OPTIONALLY supplied by the client.

4342 A client MUST be able to accept any of the attribute syntaxes defined in Section 4.1, including their full
4343 range, that may be returned to it in a response from a Printer object. In particular for each attribute that
4344 the client supports whose attribute syntax is 'text', the client MUST accept and process both the
4345 'textWithoutLanguage' and 'textWithLanguage' forms. Similarly, for each attribute that the client
4346 supports whose attribute syntax is 'name', the client MUST accept and process both the
4347 'nameWithoutLanguage' and 'nameWithLanguage' forms. For presentation purposes, truncation of long
4348 attribute values is not recommended. A recommended approach would be for the client implementation
4349 to allow the user to scroll through long attribute values.

4350 A ~~query~~-response ~~may~~MAY contain attribute groups, attributes, and values that the client does not
4351 expect. Therefore, a client implementation MUST gracefully handle such responses and not refuse to
4352 inter-operate with a conforming Printer that is returning ~~extended~~-registered or private extensions,
4353 including attribute groups, attributes, ~~and/or~~ attribute values that conform to Section 6. Clients may
4354 choose to ignore any parameters, attributes, or values that they do not understand. Issue 25 and Issue 26

4355 While a client is sending data to a printer, it SHOULD do its best to prevent a channel from being closed
4356 by a lower layer when the channel is blocked (i.e. flow-controlled off) for whatever reason, e.g. 'out of
4357 paper' or 'job ahead hasn't freed up enough memory'. However, the layer that launched the print
4358 submission (e.g. an end user) MAY close the channel in order to cancel the job. When a client closes a
4359 channel, a Printer MAY print all or part of the received portion of the document. See the "Encoding and
4360 Transport" document [IPP-PRO] for more details. Issue 4 and Issue 5

4361 A client MUST/SHOULD [which is to be determined in consultation with the Area Director] support
 4362 Client Authentication as defined in the IPP/1.1 Encoding and Transport document [IPP-PRO]. A client
 4363 SHOULD support Operation Privacy and Server Authentication as defined in the IPP/1.1 Encoding and
 4364 Transport document [IPP-PRO]. See also [IPP-MOD] section 8. Issue 32

4365 5.2 IPP Object Conformance Requirements

4366 This section specifies the conformance requirements for conforming implementations with respect to
 4367 objects, operations, and attributes whether they be (1) IPP objects that accept IPP requests and control
 4368 one or more devices or are embedded in a single device or (2) servers that accept IPP requests and
 4369 forward them to networked devices (using IPP or other protocol).

4370 5.2.1 Objects

4371 Conforming implementations MUST implement all of the model objects as defined in this specification
 4372 in the indicated sections:

4373 Section 2.1 - Printer Object

4374 Section 2.2 - Job Object

4375 whether they are (embedded) software that controls a device or are part of a print server that accepts IPP
 4376 operation requests and, in turn, sends operation requests using (the IPP or other) protocol to one or more
 4377 networked device(s). See sections 2.1 and 2.2.

4378 5.2.2 Operations

4379 Conforming IPP object implementations MUST implement all of the REQUIRED model operations,
 4380 including REQUIRED responses, as defined in this specification in the indicated sections:

4381 For a Printer object:

4382	Print-Job (section 3.2.1)	REQUIRED
4383	Print-URI (section 3.2.2)	OPTIONAL
4384	Validate-Job (section 3.2.3)	REQUIRED
4385	Create-Job (section 3.2.4)	OPTIONAL
4386	Get-Printer-Attributes (section 3.2.5)	REQUIRED
4387	Get-Jobs (section 3.2.6)	REQUIRED
4388	Pause-Printer (section 3.2.7)	OPTIONAL
4389	Resume-Printer (section 3.2.8)	OPTIONAL
4390	Purge-Jobs (section 3.2.9)	OPTIONAL

4391

4392 For a Job object:

4393	Send-Document (section 3.3.1)	OPTIONAL
4394	Send-URI (section 3.3.2)	OPTIONAL
4395	Cancel-Job (section 3.3.3)	REQUIRED
4396	Get-Job-Attributes (section 3.3.4)	REQUIRED
4397	Hold-Job (section 3.3.5)	OPTIONAL

4398 Release-Job (section 3.3.6) OPTIONAL
4399 Restart-Job (section 3.3.7) OPTIONAL

4400

4401 Conforming IPP objects MUST support all REQUIRED operation attributes and all values of such
4402 attributes if so indicated in the description. Conforming IPP objects MUST ignore all unsupported or
4403 unknown operation attributes or operation attribute groups received in a request, but MUST reject a
4404 request that contains a supported operation attribute that contains an unsupported value.

4405 Conforming IPP objects MAY return operation responses that contain attributes groups, attributes name
4406 and attribute values that are extensions to this standard. The additional attribute groups MAY occur in
4407 any order. Issue 26

4408 The following section on object attributes specifies the support required for object attributes.

4409 5.2.3 IPP Object Attributes

4410 Conforming IPP objects MUST support all of the REQUIRED object attributes, as defined in this
4411 specification in the indicated sections.

4412 If an object supports an attribute, it MUST support only those values specified in this document or
4413 through the extension mechanism described in section 5.2.4. It MAY support any non-empty subset of
4414 these values. That is, it MUST support at least one of the specified values and at most all of them.

4415 5.2.4 Versions

4416 Clients MUST support version 1.1 and ~~MAY SHOULD~~ also support version 1.0. IPP objects MUST
4417 support ~~both~~ version 1.1 and SHOULD also support version 1.0. See section 3.1.8. **ISSUE 36**

4418 5.2.5 Extensions

4419 A conforming IPP object MAY support registered extensions and private extensions, as long as they
4420 meet the requirements specified in Section 6.

4421 For each attribute included in an operation response, a conforming IPP object MUST return a value
4422 whose type and value syntax conforms to the requirement of the Model document as specified in
4423 Sections 3 and 4.

4424 5.2.6 Attribute Syntaxes

4425 An IPP object MUST be able to accept any of the attribute syntaxes defined in Section 4.1, including
4426 their full range, in any operation in which a client may supply attributes or the system administrator may
4427 configure attributes (by means outside the scope of this IPP/1.1 document). In particular for each
4428 attribute that the IPP object supports whose attribute syntax is 'text', the IPP object MUST accept and
4429 process both the 'textWithoutLanguage' and 'textWithLanguage' forms. Similarly, for each attribute that
4430 the IPP object supports whose attribute syntax is 'name', the IPP object MUST accept and process both

4431 the 'nameWithoutLanguage' and 'nameWithLanguage' forms. Furthermore, an IPP object MUST return
4432 attributes to the client in operation responses that conform to the syntax specified in Section 4.1,
4433 including their full range if supplied previously by a client.

4434 5.2.7 Security Issue 32

4435 An IPP Printer implementation MUST/SHOULD [which is to be determined in consultation with the
4436 Area Director] contain support for Client Authentication as defined in the IPP/1.1 Encoding and
4437 Transport document [IPP-PRO]. A Printer implementation MAY allow an administrator to configure
4438 the Printer so that all, some, or none of the users are authenticated. See also [IPP-MOD] section 8.

4439 An IPP Printer implementation SHOULD contain support for Operation Privacy and Server
4440 Authentication as defined in the IPP/1.1 Encoding and Transport document [IPP-PRO]. A Printer
4441 implementation MAY allow an administrator to configure the degree of support for Operation Privacy
4442 and Server Authentication. See also [IPP-MOD] section 8.

4443 5.3 Charset and Natural Language Requirements

4444 All clients and IPP objects MUST support the 'utf-8' charset as defined in section 4.1.7.

4445 IPP objects MUST be able to accept any client request which correctly uses the "attributes-natural-
4446 language" operation attribute or the Natural Language Override mechanism on any individual attribute
4447 whether or not the natural language is supported by the IPP object. If an IPP object supports a natural
4448 language, then it MUST be able to translate (perhaps by table lookup) all generated 'text' or 'name'
4449 attribute values into one of the supported languages (see section 3.1.4). That is, the IPP object that
4450 supports a natural language NEED NOT be a general purpose translator of any arbitrary 'text' or 'name'
4451 value supplied by the client into that natural language. However, the object MUST be able to translate
4452 (automatically generate) any of its own attribute values and messages into that natural language.

4453 ~~5.4 Security Conformance Requirements~~

4454 ~~Conforming IPP Printer objects SHOULD support Transport Layer Security (TLS) protocol Version 1~~
4455 ~~(TLS) [RFC2246] access, MAY support access without TLS, or MAY support both means of access.~~

4456 ~~Conforming IPP clients SHOULD support TLS access and non-TLS access. Note: This client~~
4457 ~~recommendation to support both means that conforming IPP clients will be able to inter-operate with any~~
4458 ~~IPP Printer object.~~

4459 ~~For a detailed discussion of security considerations and the IPP application security profile required for~~
4460 ~~TLS support, see section 8.~~

4461 6. IANA Considerations (registered and private extensions)

4462 This section describes how IPP can be extended to allow the following registered and private extensions
4463 to IPP:

- 4464 1. keyword attribute values
- 4465 2. enum attribute values
- 4466 3. attributes
- 4467 4. attribute syntaxes
- 4468 5. operations
- 4469 6. attribute groups
- 4470 7. status codes

4471

4472 Extensions registered for use with IPP/1.1 are OPTIONAL for client and IPP object conformance to the
4473 IPP/1.1 Model specification.

4474 These extension procedures are aligned with the guidelines as set forth by the IESG [IANA-CON].
4475 Section 11 describes how to propose new registrations for consideration. IANA will reject registration
4476 proposals that leave out required information or do not follow the appropriate format described in
4477 Section 11. IPP/1.1 may also be extended by an appropriate RFC that specifies any of the above
4478 extensions.

4479 6.1 Typed 'keyword' and 'enum' Extensions

4480 IPP allows for 'keyword' and 'enum' extensions (see sections 4.1.2.3 and 4.1.4). This document uses
4481 prefixes to the 'keyword' and 'enum' basic attribute syntax type in order to communicate extra
4482 information to the reader through its name. This extra information is not represented in the protocol
4483 because it is unimportant to a client or Printer object. The list below describes the prefixes and their
4484 meaning.

4485 "type1": The IPP specification must be revised to add a new keyword or a new enum. No private
4486 keywords or enums are allowed.

4487

4488 "type2": Implementers can, at any time, add new keyword or enum values by proposing the
4489 complete specification to IANA:

4490

4491 iana@iana.org

4492

4493 IANA will forward the registration proposal to the IPP Designated Expert who will review the
4494 proposal with a mailing list that the Designated Expert keeps for this purpose. Initially, that list
4495 will be the mailing list used by the IPP WG:

4496

4497 ipp@pwg.org

4498

4499 even after the IPP WG is disbanded as permitted by [IANA-CON]. The IPP Designated Expert
4500 is appointed by the IESG Area Director responsible for IPP, according to [IANA-CON].

4501
4502 When a type2 keyword or enum is approved, the IPP Designated Expert becomes the point of
4503 contact for any future maintenance that might be required for that registration.

4504
4505 "type3": Implementers can, at any time, add new keyword and enum values by submitting the
4506 complete specification to IANA as for type2 who will forward the proposal to the IPP Designated
4507 Expert. While no additional technical review is required, the IPP Designated Expert may, at
4508 his/her discretion, forward the proposal to the same mailing list as for type2 registrations for
4509 advice and comment.

4510
4511 When a type3 keyword or enum is approved by the IPP Designated Expert, the original proposer
4512 becomes the point of contact for any future maintenance that might be required for that
4513 registration.

4514
4515 For type2 and type3 keywords, the proposer includes the name of the keyword in the registration
4516 proposal and the name is part of the technical review.

4517 After type2 and type3 enums specifications are approved, the IPP Designated Expert in consultation with
4518 IANA assigns the next available enum number for each enum value.

4519 IANA will publish approved type2 and type3 keyword and enum attributes value registration
4520 specifications in:

4521 `ftp.isi.edu/iana/assignments/ipp/attribute-values/xxx/yyy.txt`

4522 where xxx is the attribute name that specifies the initial values and yyy.txt is a descriptive file name that
4523 contains one or more enums or keywords approved at the same time. For example, if several additional
4524 enums for stapling are approved for use with the "finishings" attribute (and "finishings-default" and
4525 "finishings-supported" attributes), IANA will publish the additional values in the file:

4526 `ftp.isi.edu/iana/assignments/ipp/attribute-values/finishings/stapling.txt`

4527 Note: Some attributes are defined to be: 'type3 keywords' | 'name' which allows for attribute values to be
4528 extended by a site administrator with administrator defined names. Such names are not registered with
4529 IANA.

4530 By definition, each of the three types above assert some sort of registry or review process in order for
4531 extensions to be considered valid. Each higher numbered level (1, 2, 3) tends to be decreasingly less
4532 stringent than the previous level. Therefore, any typeN value MAY be registered using a process for
4533 some typeM where M is less than N, however such registration is NOT REQUIRED. For example, a
4534 type3 value MAY be registered in a type 1 manner (by being included in a future version of an IPP
4535 specification), however, it is NOT REQUIRED.

4536 This specification defines keyword and enum values for all of the above types, including type3
4537 keywords.

4538 For private (unregistered) keyword extensions, implementers SHOULD use keywords with a suitable
4539 distinguishing prefix, such as "xxx-" where xxx is the (lowercase) fully qualified company name
4540 registered with IANA for use in domain names [RFC1035]. For example, if the company XYZ Corp.
4541 had obtained the domain name "XYZ.com", then a private keyword 'abc' would be: 'xyz.com-abc'.

4542 Note: RFC 1035 [RFC1035] indicates that while upper and lower case letters are allowed in domain
4543 names, no significance is attached to the case. That is, two names with the same spelling but different
4544 case are to be treated as if identical. Also, the labels in a domain name must follow the rules for
4545 ARPANET host names: They must start with a letter, end with a letter or digit, and have as interior
4546 characters only letters, digits, and hyphen. Labels must be 63 characters or less. Labels are separated by
4547 the "." character.

4548 For private (unregistered) enum extension, implementers MUST use values in the reserved integer range
4549 which is 2**30 to 2**31-1.

4550 6.2 Attribute Extensibility

4551 Attribute names are type2 keywords. Therefore, new attributes may be registered and have the same
4552 status as attributes in this document by following the type2 extension rules. For private (unregistered)
4553 attribute extensions, implementers SHOULD use keywords with a suitable distinguishing prefix as
4554 described in Section 6.1.

4555 IANA will publish approved attribute registration specifications as separate files:

4556 ftp.isi.edu/iana/assignments/ipp/attributes/xxx-yyy.txt

4557 where "xxx-yyy" is the new attribute name.

4558 If a new Printer object attribute is defined and its values can be affected by a specific document format,
4559 its specification needs to contain the following sentence:

4560 "The value of this attribute returned in a Get-Printer-Attributes response MAY depend on the
4561 "document-format" attribute supplied (see Section 3.2.5.1)."

4562 If the specification does not, then its value in the Get-Printer-Attributes response MUST NOT depend on
4563 the "document-format" supplied in the request. When a new Job Template attribute is registered, the
4564 value of the Printer attributes MAY vary with "document-format" supplied in the request without the
4565 specification having to indicate so.

4566 6.3 Attribute Syntax Extensibility

4567 Attribute syntaxes are like type2 enums. Therefore, new attribute syntaxes may be registered and have
4568 the same status as attribute syntaxes in this document by following the type2 extension rules described in

4569 Section 6.1. The value codes that identify each of the attribute syntaxes are assigned in the "Encoding
4570 and Transport" specification [IPP-PRO], including a designated range for private, experimental use.

4571 For attribute syntaxes, the IPP Designated Expert in consultation with IANA assigns the next attribute
4572 syntax code in the appropriate range as specified in [IPP-PRO]. IANA will publish approved attribute
4573 syntax registration specifications as separate files:

4574 `ftp.isi.edu/iana/assignments/ipp/attribute-syntaxes/xxx-yyy.txt`

4575 where 'xxx-yyy' is the new attribute syntax name.

4576 6.4 Operation Extensibility

4577 Operations may also be registered following the type2 procedures described in Section 6.1, though major
4578 new operations will usually be done by a new standards track RFC that augments this document. For
4579 private (unregistered) operation extensions, implementers MUST use the range for the "operation-id" in
4580 requests specified in Section 4.4.15 "operations-supported" Printer attribute.

4581 For operations, the IPP Designated Expert in consultation with IANA assigns the next operation-id code
4582 as specified in Section 4.4.15. IANA will publish approved operation registration specifications as
4583 separate files:

4584 `ftp.isi.edu/iana/assignments/ipp/operations/Xxx-Yyy.txt`

4585 where "Xxx-Yyy" is the new operation name.

4586 6.5 Attribute Groups

4587 Attribute groups passed in requests and responses may be registered following the type2 procedures
4588 described in Section 6.1. The tags that identify each of the attribute groups are assigned in [IPP-PRO].

4589 For attribute groups, the IPP Designated Expert in consultation with IANA assigns the next attribute
4590 group tag code in the appropriate range as specified in [IPP-PRO]. IANA will publish approved
4591 attribute group registration specifications as separate files:

4592 `ftp.isi.edu/iana/assignments/ipp/attribute-group-tags/xxx-yyy-tag.txt`

4593 where 'xxx-yyy-tag' is the new attribute group tag name.

4594 6.6 Status Code Extensibility

4595 Operation status codes may also be registered following the type2 procedures described in Section 6.1.
4596 The values for status codes are allocated in ranges as specified in Section 14 for each status code class:

4597 "informational" - Request received, continuing process

4598 "successful" - The action was successfully received, understood, and accepted

4599 "redirection" - Further action must be taken in order to complete the request

4600 "client-error" - The request contains bad syntax or cannot be fulfilled

4601 "server-error" - The IPP object failed to fulfill an apparently valid request

4602

4603 For private (unregistered) operation status code extensions, implementers MUST use the top of each
4604 range as specified in Section 13.

4605 For operation status codes, the IPP Designated Expert in consultation with IANA assigns the next status
4606 code in the appropriate class range as specified in Section 13. IANA will publish approved status code
4607 registration specifications as separate files:

4608 ftp.isi.edu/iana/assignments/ipp/status-codes/xxx-yyy.txt

4609 where "xxx-yyy" is the new operation status code keyword.

4610 6.7 Registration of MIME types/sub-types for document-formats

4611 The "document-format" attribute's syntax is 'mimeMediaType'. This means that valid values are Internet
4612 Media Types (see Section 4.1.9). RFC 2045 [RFC2045] defines the syntax for valid Internet media
4613 types. IANA is the registry for all Internet media types.

4614 6.8 Registration of charsets for use in 'charset' attribute values

4615 The "attributes-charset" attribute's syntax is 'charset'. This means that valid values are charsets names.
4616 When a charset in the IANA registry has more than one name (alias), the name labeled as "(preferred
4617 MIME name)", if present, MUST be used (see Section 4.1.7). IANA is the registry for charsets
4618 following the procedures of [RFC2278].

4619 7. Internationalization Considerations

4620 Some of the attributes have values that are text strings and names which are intended for human
4621 understanding rather than machine understanding (see the 'text' and 'name' attribute syntaxes in Sections
4622 4.1.1 and 4.1.2).

4623 In each operation request, the client

4624 - identifies the charset and natural language of the request which affects each supplied 'text' and
4625 'name' attribute value, and

4626 - requests the charset and natural language for attributes returned by the IPP object in operation
4627 responses (as described in Section 3.1.4.1).

4628

4629 In addition, the client MAY separately and individually identify the Natural Language Override of a
4630 supplied 'text' or 'name' attribute using the 'textWithLanguage' and 'nameWithLanguage' technique
4631 described section 4.1.1.2 and 4.1.2.2 respectively.

4632 All IPP objects MUST support the UTF-8 [RFC2279] charset in all 'text' and 'name' attributes supported.
4633 If an IPP object supports more than the UTF-8 charset, the object MUST convert between them in order
4634 to return the requested charset to the client according to Section 3.1.4.2. If an IPP object supports more
4635 than one natural language, the object SHOULD return 'text' and 'name' values in the natural language
4636 requested where those values are generated by the Printer (see Section 3.1.4.1).

4637 For Printers that support multiple charsets and/or multiple natural languages in 'text' and 'name'
4638 attributes, different jobs may have been submitted in differing charsets and/or natural languages. All
4639 responses MUST be returned in the charset requested by the client. However, the Get-Jobs operation
4640 uses the 'textWithLanguage' and 'nameWithLanguage' mechanism to identify the differing natural
4641 languages with each job attribute returned.

4642 The Printer object also has configured charset and natural language attributes. The client can query the
4643 Printer object to determine the list of charsets and natural languages supported by the Printer object and
4644 what the Printer object's configured values are. See the "charset-configured", "charset-supported",
4645 "natural-language-configured", and "generated-natural-language-supported" Printer description attributes
4646 for more details.

4647 The "charset-supported" attributed identifies the supported charsets. If a charset is supported, the IPP
4648 object MUST be capable of converting to and from that charset into any other supported charset. In
4649 many cases, an IPP object will support only one charset and it MUST be the UTF-8 charset.

4650 The "charset-configured" attribute identifies the one supported charset which is the native charset given
4651 the current configuration of the IPP object (administrator defined).

4652 The "generated-natural-language-supported" attribute identifies the set of supported natural languages
4653 for generated messages; it is not related to the set of natural languages that must be accepted for client
4654 supplied 'text' and 'name' attributes. For client supplied 'text' and 'name' attributes, an IPP object MUST
4655 accept ALL supplied natural languages. Just because a Printer object is currently configured to support
4656 'en-us' natural language does not mean that the Printer object should reject a job if the client supplies a
4657 job name that is in 'fr-ca'.

4658 The "natural-language-configured" attribute identifies the one supported natural language for generated
4659 messages which is the native natural language given the current configuration of the IPP object
4660 (administrator defined).

4661 Attributes of type 'text' and 'name' are populated from different sources. These attributes can be
4662 categorized into following groups (depending on the source of the attribute):

- 4663 1. Some attributes are supplied by the client (e.g., the client supplied "job-name", "document-name",
4664 and "requesting-user-name" operation attributes along with the corresponding Job object's "job-
4665 name" and "job-originating-user-name" attributes). The IPP object MUST accept these attributes
4666 in any natural language no matter what the set of supported languages for generated messages
- 4667 2. Some attributes are supplied by the system administrator (e.g., the Printer object's "printer-name"
4668 and "printer-location" attributes). These too can be in any natural language. If the natural
4669 language for these attributes is different than what a client requests, then they must be reported
4670 using the Natural Language Override mechanism.

- 4671 3. Some attributes are supplied by the device manufacturer (e.g., the Printer object's "printer-make-
 4672 and-model" attribute). These too can be in any natural language. If the natural language for
 4673 these attributes is different than what a client requests, then they must be reported using the
 4674 Natural Language Override mechanism.
- 4675 4. Some attributes are supplied by the operator (e.g., the Job object's "job-message-from-operator"
 4676 attribute). These too can be in any natural language. If the natural language for these attributes is
 4677 different than what a client requests, then they must be reported using the Natural Language
 4678 Override mechanism.
- 4679 5. Some attributes are generated by the IPP object (e.g., the Job object's "job-state-message"
 4680 attribute, the Printer object's "printer-state-message" attribute, and the "status-message" operation
 4681 attribute). These attributes can only be in one of the "generated-natural-language-supported"
 4682 natural languages. If a client requests some natural language for these attributes other than one of
 4683 the supported values, the IPP object SHOULD respond using the value of the "natural-language-
 4684 configured" attribute (using the Natural Language Override mechanism if needed).

4686 The 'text' and 'name' attributes specified in this version of this document (additional ones will be
 4687 registered according to the procedures in Section 6) are:

Attributes	Source
Operation Attributes:	
job-name (name)	client
document-name (name)	client
requesting-user-name (name)	client
status-message	Job or Printer object
Job Template Attributes:	
job-hold-until (keyword name)	client matches administrator-configured
job-hold-until-default (keyword name)	client matches administrator-configured
job-hold-until-supported (keyword name)	client matches administrator-configured
job-sheets (keyword name)	client matches administrator-configured
job-sheets-default (keyword name)	client matches administrator-configured
job-sheets-supported (keyword name)	client matches administrator-configured
media (keyword name)	client matches administrator-configured
media-default (keyword name)	client matches administrator-configured
media-supported (keyword name)	client matches administrator-configured
media-ready (keyword name)	client matches administrator-configured
Job Description Attributes:	
job-name (name)	client or Printer object
job-originating-user-name (name)	Printer object
job-state-message (text)	Job or Printer object
output-device-assigned (name(127))	administrator
job-message-from-operator (text(127))	operator

Printer	Description Attributes:	
	printer-name (name(127))	administrator
	printer-location (text(127))	administrator
	printer-info (text(127))	administrator
	printer-make-and-model (text(127))	administrator or manufacturer
	printer-state-message (text)	Printer object
	printer-message-from-operator (text(127))	operator

4688 8. Security Considerations

4689 ~~IPP objects SHOULD be deployed over protocol stacks that support the Transport Layer Security (TLS)~~
4690 ~~protocol [RFC2246]. Other IPP objects MAY be deployed over protocol stacks that do not support TLS.~~
4691 ~~Some IPP objects MAY be deployed over both types of protocol stacks. Those IPP objects that support~~
4692 ~~TLS, are capable of supporting mutual authentication as well as privacy of messages via multiple~~
4693 ~~encryption schemes. An important point about security related information for TLS access to an IPP~~
4694 ~~object, is that the security related parameters (authentication, encryption keys, etc.) are "out of band" to~~
4695 ~~the actual IPP protocol.~~

4696 ~~An IPP object that does not support TLS MAY elect to support a transport layer that provides other~~
4697 ~~security mechanisms. For example, in a mapping of IPP over HTTP/1.1 [IPP-PRO], if the IPP object~~
4698 ~~does not support TLS, HTTP still allows for client authentication using Digest Access Authentication~~
4699 ~~(DAA) [RFC2069].~~

4700 It is difficult to anticipate the security risks that might exist in any given IPP environment. For example,
4701 if IPP is used within a given corporation over a private network, the risks of exposing document data
4702 may be low enough that the corporation will choose not to use encryption on that data. However, if the
4703 connection between the client and the IPP object is over a public network, the client may wish to protect
4704 the content of the information during transmission through the network with encryption.

4705 Furthermore, the value of the information being printed may vary from one IPP environment to the next.
4706 Printing payroll checks, for example, would have a different value than printing public information from
4707 a file. There is also the possibility of denial-of-service attacks, but denial-of-service attacks against
4708 printing resources are not well understood and there is no published precedents regarding this scenario.

4709 Once the authenticated identity of the requester has been supplied to the IPP object, the object uses that
4710 identity to enforce any authorization policy that might be in place. For example, one site's policy might
4711 be that only the job owner is allowed to cancel a job. The details and mechanisms to set up a particular
4712 access control policy are not part of IPP/1.1, and must be established via some other type of
4713 administrative or access control framework. However, there are operation status codes that allow an IPP
4714 server to return information back to a client about any potential access control violations for an IPP
4715 object.

4716 During a create operation, the client's identity is recorded in the Job object in an implementation-defined
4717 attribute. This information can be used to verify a client's identity for subsequent operations on that Job

4718 object in order to enforce any access control policy that might be in effect. See section 8.3 below for
4719 more details.

4720 Since the security levels or the specific threats that any given IPP system administrator may be
4721 concerned with cannot be anticipated, IPP MUST be capable of operating with different security
4722 mechanisms and security policies as required by the individual installation. Security policies might vary
4723 from very strong, to very weak, to none at all, and corresponding security mechanisms will be required.
4724 ~~TLS supports the type of negotiated levels of security required by most, if not all, potential IPP~~
4725 ~~environments. IPP environments that require no security can elect to deploy IPP objects that do not~~
4726 ~~utilize the optional TLS security mechanisms.~~

4727 8.1 Security Scenarios

4728 The following sections describe specific security attacks for IPP environments. Where examples are
4729 provided they should be considered illustrative of the environment and not an exhaustive set. Not all of
4730 these environments will necessarily be addressed in initial implementations of IPP.

4731 8.1.1 Client and Server in the Same Security Domain

4732 This environment is typical of internal networks where traditional office workers print the output of
4733 personal productivity applications on shared work-group printers, or where batch applications print their
4734 output on large production printers. Although the identity of the user may be trusted in this environment,
4735 a user might want to protect the content of a document against such attacks as eavesdropping, replaying
4736 or tampering.

4737 8.1.2 Client and Server in Different Security Domains

4738 Examples of this environment include printing a document created by the client on a publicly available
4739 printer, such as at a commercial print shop; or printing a document remotely on a business associate's
4740 printer. This latter operation is functionally equivalent to sending the document to the business associate
4741 as a facsimile. Printing sensitive information on a Printer in a different security domain requires strong
4742 security measures. In this environment authentication of the printer is required as well as protection
4743 against unauthorized use of print resources. Since the document crosses security domains, protection
4744 against eavesdropping and document tampering are also required. It will also be important in this
4745 environment to protect Printers against "spamming" and malicious document content.

4746 8.1.3 Print by Reference

4747 When the document is not stored on the client, printing can be done by reference. That is, the print
4748 request can contain a reference, or pointer, to the document instead of the actual document itself (see
4749 [sections 3.2.2 and 3.3.2](#)). Standard methods currently do not exist for remote entities to "assume" the
4750 credentials of a client for forwarding requests to a 3rd party. It is anticipated that Print-By-Reference will
4751 be used to access "public" documents and that sophisticated methods for authenticating "proxies" ~~will~~
4752 ~~not be~~ ~~specified~~ ~~for version 1 of IPP~~ ~~in this document~~.

4753 8.2 URIs in Operation, Job, and Printer attributes~~for TLS and non-TLS Access~~

4754 ~~As described earlier, an IPP object SHOULD support TLS access, MAY non-TLS access, or both.~~ The
4755 "printer-uri-supported" attribute contains the Printer object's URI(s). Its companion attribute, "uri-
4756 security-supported", identifies the security mechanism used for each URI listed in the "printer-uri-
4757 supported" attribute. For each Printer operation request, a client MUST supply only one URI in the
4758 "printer-uri" operation attribute. In other words, even though the Printer supports more than one URI,
4759 the client only interacts with the Printer object using one of its URIs. This duality is not needed for Job
4760 objects, since the Printer object is the factory for Job objects, and the Printer object will generate the
4761 correct URI for new Job objects depending on the Printer object's security configuration.

4762 8.3 URIs for each authentication mechanisms~~The "requesting-user-name" (name(MAX)) Operation~~
4763 ~~Attribute~~

4764 Each URI has an authentication mechanism associated with it. If the URI is the ith element of "printer-
4765 uri-supported", then authentication mechanism is the "ith" element of "uri-authentication-supported".
4766 For a list of possible authentication mechanisms, see section 4.4.2.

4767 The Printer object uses an authentication mechanism to determine the name of the user performing an
4768 operation. This user is called the "authenticated user". The credibility of authentication depends on the
4769 mechanism that the Printer uses to obtain the user's name. When the authentication mechanism is 'none',
4770 all authenticated users are "anonymous".

4771 During job creation operations, the Printer initializes the value of the "job-originating-user-name"
4772 attribute to be the authenticated user. The authenticated user in this case is called the "job-owner".

4773 If an implementation can be configured to support more than one authentication mechanism, then it
4774 MUST implement rules for determining equality of authenticated user names which have been
4775 authenticated via different authentication mechanisms. One possible policy is that identical names that
4776 are authenticated via different mechanism are different. For example, a user can cancel his job only if he
4777 uses the same authentication mechanism for both Cancel-Job and Print-Job. Another policy is that
4778 identical names that are authenticated via different mechanism are the same if the authentication
4779 mechanism for the later operation is not less strong than the authentication mechanism for the earlier job
4780 creation operation. For example, a user can cancel his job only if he uses the same or stronger
4781 authentication mechanism for Cancel-Job and Print-Job. With this second policy a job submitted via
4782 'requesting-user-name' authentication could be cancelled via 'digest' authentication. With the first policy,
4783 the job could not be cancelled in this way.

4784 A client is able to determine the authentication mechanism used to create a job. It is the ith value of the
4785 Printer's "uri-authentication-supported" attribute, where i is the index of the element of the Printer's
4786 "uri-printer-supported" attribute equal to the job's "job-printer-uri" attribute.

4787 ~~Each operation MUST specify the user who is performing the operation in both of the following two~~
4788 ~~ways:~~

- 4789 ~~1) via the REQUIRED "requesting-user-name" operation attribute that a client SHOULD supply in~~
4790 ~~all operations. The client MUST obtain the value for this attribute from an environmental or~~
4791 ~~network login name for the user, rather than allowing the user to supply any value. If the client~~
4792 ~~does not supply a value for "requesting-user-name", the printer MUST assume that the client is~~
4793 ~~supplying some anonymous name, such as "anonymous".~~
4794 ~~2) via an authentication mechanism of the underlying transport which may be configured to give no~~
4795 ~~authentication information.~~

4797 ~~There are six cases to consider:~~

- 4798 ~~a) the authentication mechanism gives no information, and the client doesn't specify "requesting-~~
4799 ~~user-name".~~
4800 ~~b) the authentication mechanism gives no information, but the client specifies "requesting-user-~~
4801 ~~name".~~
4802 ~~c) the authentication mechanism specifies a user which has no human readable representation, and~~
4803 ~~the client doesn't specify "requesting-user-name".~~
4804 ~~d) the authentication mechanism specifies a user which has no human readable representation, but~~
4805 ~~the client specifies "requesting-user-name".~~
4806 ~~e) the authentication mechanism specifies a user which has a human readable representation. The~~
4807 ~~Printer object ignores the "requesting-user-name".~~
4808 ~~f) the authentication mechanism specifies a user who is trusted and whose name means that the~~
4809 ~~value of the "requesting-user-name", which MUST be present, is treated as the authenticated~~
4810 ~~name.~~

4812 ~~Note: Case "f" is intended for a tightly coupled gateway and server to work together so that the "user"~~
4813 ~~name is able to be that of the gateway client and not that of the gateway. Because most, if not all, system~~
4814 ~~vendors will initially implement IPP via a gateway into their existing print system, this mechanism is~~
4815 ~~necessary unless the authentication mechanism allows a gateway (client) to act on behalf of some other~~
4816 ~~client.~~

4817 ~~The user name has two forms:~~

- 4818 ~~—one that is human readable: it is held in the REQUIRED "job-originating-user-name" Job~~
4819 ~~Description attribute which is set during the job creation operations. It is used for presentation~~
4820 ~~only, such as returning in queries or printing on start sheets~~
4821 ~~—one for authorization: it is held in an undefined (by IPP) Job object attribute which is set by the job~~
4822 ~~creation operation. It is used to authorize other operations, such as Send Document, Send URI,~~
4823 ~~Cancel Job, to determine the user when the "my-jobs" attribute is specified with Get Jobs, and to~~
4824 ~~limit what attributes and values to return with Get Job Attributes and Get Jobs.~~

4826 ~~The human readable user name:~~

- 4827 ~~—is the value of the "requesting-user-name" for cases b, d and f.~~
4828 ~~—comes from the authentication mechanism for case e~~
4829 ~~—is some anonymous name, such as "anonymous" for cases a and c.~~

4830

4831 ~~The user name used for authorization:~~

4832 ~~—is the value of the "requesting user name" for cases b and f.~~

4833 ~~—comes from the authentication mechanism for cases c, d and e~~

4834 ~~—is some anonymous name, such as "anonymous" for case a.~~

4835

4836 ~~The essence of these rules for resolving conflicting sources of user names is that a printer~~
4837 ~~implementation is free to pick either source as long as it achieves consistent results. That is, if a user~~
4838 ~~uses the same path for a series of requests, the requests MUST appear to come from the same user from~~
4839 ~~the standpoint of both the human readable user name and the user name for authorization. This rule~~
4840 ~~MUST continue to apply even if a request could be authenticated by two or more mechanisms. It doesn't~~
4841 ~~matter which of several authentication mechanisms a Printer uses as long as it achieves consistent~~
4842 ~~results. If a client uses more than one authentication mechanism, it is recommended that an~~
4843 ~~administrator make all credentials resolve to the same user and user name as much as possible.~~

4844 8.4 Restricted Queries

4845 In many IPP operations, a client supplies a list of attributes to be returned in the response. For security
4846 reasons, an IPP object may be configured not to return all attributes (or all values) that a client requests.
4847 The job attributes returned MAY depend on whether the requesting user is the same as the user that
4848 submitted the job. The IPP object MAY even return none of the requested attributes. In such cases, the
4849 status returned is the same as if the object had returned all requested attributes. The client cannot tell by
4850 such a response whether the requested attribute was present or absent on the object.

4851 8.5 Operations performed by operators and system administrators

4852 For the three printer operations Pause-Printer, Resume-Printer, and Purge-Jobs (see sections 3.2.7, 3.2.8
4853 and 3.2.9), the requesting user is intended to be an operator or administrator of the Printer object (see
4854 section 1). For operations on jobs, the requesting user is intended to be the job owner or may be an
4855 operator or administrator of the Printer object. The means for authorizing an operator or administrator of
4856 the Printer object are not specified in this document.

4857 8.6 Queries on jobs submitted using non-IPP protocols

4858 If the device that an IPP Printer is representing is able to accept jobs using other job submission
4859 protocols in addition to IPP, it is RECOMMENDED that such an implementation at least allow such
4860 "foreign" jobs to be queried using Get-Jobs returning "job-id" and "job-uri" as 'unknown'. Such an
4861 implementation NEED NOT support all of the same IPP job attributes as for IPP jobs. The IPP object
4862 returns the 'unknown' out-of-band value for any requested attribute of a foreign job that is supported for
4863 IPP jobs, but not for foreign jobs.

4864 It is further RECOMMENDED, that the IPP Printer generate "job-id" and "job-uri" values for such
4865 "foreign jobs", if possible, so that they may be targets of other IPP operations, such as Get-Job-Attributes
4866 and Cancel-Job. Such an implementation also needs to deal with the problem of authentication of such
4867 foreign jobs. One approach would be to treat all such foreign jobs as belonging to users other than the

4868 user of the IPP client. Another approach would be for the foreign job to belong to 'anonymous'. Only if
4869 the IPP client has been authenticated as an operator or administrator of the IPP Printer object, could the
4870 foreign jobs be queried by an IPP request. Alternatively, if the security policy is to allow users to query
4871 other users' jobs, then the foreign jobs would also be visible to an end-user IPP client using Get-Jobs and
4872 Get-Job-Attributes.

4873 ~~8.7 IPP Security Application Profile for TLS~~

4874 ~~The IPP application profile for TLS follows the standard "Mandatory Cipher Suites" requirement~~
4875 ~~as documented in the TLS specification [RFC2246].~~

4876 ~~If a conforming IPP object supports TLS, it MUST implement and support the "Mandatory Cipher~~
4877 ~~Suites" as specified in the TLS specification [RFC2246] and MAY support additional cipher suites.~~

4878 ~~A conforming IPP client SHOULD support TLS including the "Mandatory Cipher Suites" as specified in~~
4879 ~~the TLS specification [RFC2246]. A conforming IPP client MAY support additional cipher suites.~~
4880 ~~Client implementations MUST NOT assume any other cipher suites are supported by an IPP Printer~~
4881 ~~object.~~

4882 ~~See the TLS specification [RFC2246] for a discussion of any government export restrictions on~~
4883 ~~implementations conforming to the "Mandatory Cipher Suites".~~

4884 9. References

4885 [ASCII]

4886 Coded Character Set - 7-bit American Standard Code for Information Interchange (ASCII), ANSI
4887 X3.4-1986. This standard is the specification of the US-ASCII charset.

4888 [BCP-11]

4889 [Bradner S.](#), [Hovey R.](#), "The Organizations Involved in the IETF Standards Process", 1996/10/29
4890 (RFC 2028)

4891 [HTPP]

4892 J. Barnett, K. Carter, R. DeBry, "Initial Draft - Hypertext Printing Protocol - HTPP/1.0",
4893 October 1996, <ftp://ftp.pwg.org/pub/pwg/ipp/historic/http/overview.ps.gz>

4894 [IANA-CON]

4895 Narte, T. and Alvestrand, H.T.: Guidelines for Writing an IANA Considerations Section in
4896 RFCs, Work in Progress, draft-iesg-iana-considerations-04.txt, May 21, 1998.

4897 [IANA-CS]

4898 IANA Registry of Coded Character Sets: <ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets>

4899 [IANA-MT]

4900 IANA Registry of Media Types: <ftp://ftp.isi.edu/in-notes/iana/assignments/media-types/>

- 4901 [IPP-IIG]
4902 Hastings, T., Manros, C., "Internet Printing Protocol/1.1: draft-ietf-ipp-implementers-guide-v11-
4903 ???.txt, ?? 1999, work in progress.
- 4904 [IPP-IIG1.0]
4905 Hastings, T., Manros, C., "Internet Printing Protocol/1.0: Implementer's Guide", draft-ietf-ipp-
4906 implementers-guide-01.txt, February 1999, work in progress.
- 4907 [~~RFC2569~~IPP-LPD]
4908 Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols",
4909 draft-ietf-ipp-lpd-ipp-map-05.txt, November 1998.
- 4910 [~~IPP-MOD1.0~~RFC2566]
4911 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model
4912 and Semantics", ~~RFC 2566, April 1999~~draft-ietf-ipp-model-11.txt, ~~November, 1998~~.
- 4913 [IPP-PRO]
4914 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.1: Encoding and
4915 Transport", draft-ietf-ipp-protocol-v11-010.txt, ~~February~~May, 1999.
- 4916 [~~RFC2565~~IPP-PRO1.0]
4917 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.0: Encoding and
4918 Transport", ~~RFC 2565, April 1999~~draft-ietf-ipp-protocol-07.txt, ~~November, 1998~~.
- 4919 [~~RFC2568~~IPP-RAT]
4920 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",
4921 draft-ietf-ipp-rat-04.txt, November, 1998.
- 4922 [~~RFC2567~~IPP-REQ]
4923 Wright, D., "Design Goals for an Internet Printing Protocol", draft-ietf-ipp-req-03.txt, November,
4924 1998.
- 4925 [ISO10646-1]
4926 ISO/IEC 10646-1:1993, "Information technology -- Universal Multiple-Octet Coded Character
4927 Set (UCS) - Part 1: Architecture and Basic Multilingual Plane, JTC1/SC2."
- 4928 [ISO8859-1]
4929 ISO/IEC 8859-1:1987, "Information technology -- 8-bit One-Byte Coded Character Set - Part 1:
4930 Latin Alphabet Nr 1", 1987, JTC1/SC2.
- 4931 [ISO10175]
4932 ISO/IEC 10175 Document Printing Application (DPA), June 1996.
- 4933 [LDPA]
4934 T. Hastings, S. Isaacson, M. MacKay, C. Manros, D. Taylor, P. Zehler, "LDPA - Lightweight
4935 Document Printing Application", October 1996,
4936 ftp://ftp.pwg.org/pub/pwg/ipp/historic/ldpa/ldpa8.pdf.gz

- 4937 [P1387.4]
4938 Kirk, M. (editor), POSIX System Administration - Part 4: Printing Interfaces, POSIX 1387.4 D8,
4939 1994.
- 4940 [PSIS] Herriot, R. (editor), X/Open A Printing System Interoperability Specification (PSIS), August
4941 1995.
- 4942 [PWG]
4943 Printer Working Group, <http://www.pwg.org>.
- 4944 [RFC1035]
4945 P. Mockapetris, "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION", RFC
4946 1035, November 1987.
- 4947 [RFC1179]
4948 McLaughlin, L. III, (editor), "Line Printer Daemon Protocol" RFC 1179, August 1990.
- 4949 [RFC1759]
4950 Smith, R., Wright, F., Hastings, T., Zilles, S., and Gyllenskog, J., "Printer MIB", RFC 1759,
4951 March 1995.
- 4952 [RFC1766]
4953 H. Alvestrand, "Tags for the Identification of Languages", RFC 1766, March 1995.
- 4954 [RFC1903]
4955 J. Case, et.al., "Textual Conventions for Version 2 of the Simple Network Management Protocol
4956 (SNMP v2)" RFC 1903, January 1996.
- 4957 [RFC1952]
4958 P. Deutsch, "GZIP file format specification version 4.3", RFC 1952, May 1996.
- 4959 [RFC2026]
4960 S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.
- 4961 [RFC2045]
4962 N. Fried, N. Borenstein, ", Multipurpose Internet Mail Extensions (MIME) Part One: Format of
4963 Internet Message Bodies " RFC 2045, November 1996.
- 4964 [RFC2046]
4965 Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. N. Freed & N.
4966 Borenstein. November 1996. (Obsoletes RFC1521, RFC1522, RFC1590), RFC 2046.
- 4967 [RFC2048]
4968 N. Freed, J. Klensin & J. Postel, "Multipurpose Internet Mail Extension (MIME) Part Four:
4969 Registration Procedures". RFC 2048, November 1996.

- 4970 [RFC2068]
4971 R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, "Hypertext Transfer Protocol -
4972 HTTP/1.1", RFC 2068, January 1997
- 4973 [RFC2069]
4974 J. Franks, P. Hallam-Baker, J. Hostetler, P. Leach, A. Luotonen, E. Sink, L. Stewart, "An
4975 Extension to HTTP: Digest Access Authentication", RFC-2069, Jan 1997.
- 4976 [RFC2119]
4977 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 , March
4978 1997
- 4979 [RFC2228]
4980 M. Horowitz, S. Lunt, "FTP Security Extensions", RFC 2228, October 1997.
- 4981 [RFC2246]
4982 T. Dierks, C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.
- 4983 [RFC2277]
4984 H. Alvestrand, "IETF Policy on Character Sets and Languages" RFC 2277, January 1998.
- 4985 [RFC2278]
4986 N. Freed, J. Postel: "IANA CharSet Registration Procedures", RFC 2278, January 1998.
- 4987 [RFC2279]
4988 F. Yergeau , "UTF-8, a transformation format of ISO 10646", RFC 2279. January 1998.
- 4989 [RFC2316]
4990 S. Bellovin , "Report of the IAB Security Architecture Workshop", RFC 2316, April 1998.
- 4991 [RFC2396]
4992 Berners-Lee, T., Fielding, R., Masinter, L., "Uniform Resource Identifiers (URI): Generic
4993 Syntax", RFC 2396, August 1998.
- 4994 [SSL]
4995 Netscape, The SSL Protocol, Version 3, (Text version 3.02), November 1996.
- 4996 [SWP]
4997 P. Moore, B. Jahromi, S. Butler, "Simple Web Printing SWP/1.0", May 7, 1997,
4998 ftp://ftp.pwg.org/pub/pwg/ipp/new_PRO/swp9705.pdf

4999 ~~10. Notices~~

5000 ~~The IETF takes no position regarding the validity or scope of any intellectual property or other rights that~~
5001 ~~might be claimed to pertain to the implementation or use of the technology described in this document~~
5002 ~~or the extent to which any license under such rights might or might not be available; neither does it~~
5003 ~~represent that it has made any effort to identify any such rights. Information on the IETF's procedures~~
5004 ~~with respect to rights in standards track and standards-related documentation can be found in BCP-~~

~~11[BCP-11]. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF Secretariat.~~

~~The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.~~

~~Copyright (C) The Internet Society (1999). All Rights Reserved.~~

~~This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.~~

~~The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.~~

~~This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.~~

10. Author's Address

Scott A. Isaacson (Editor)
Novell, Inc.
122 E 1700 S
Provo, UT 84606

Phone: 801-861-7366
Fax: 801-861-2517
e-mail: sisaacson@novell.com

Tom Hastings
Xerox Corporation
737 Hawaii St. ESAE 231
El Segundo, CA 90245

5043
5044 Phone: 310-333-6413
5045 Fax: 310-333-5514
5046 e-mail: hastings@cp10.es.xerox.com

5047
5048 Robert Herriot
5049 Xerox Corp.
5050 3400 Hill View Ave, Building 1
5051 Palo Alto, CA 94304

5052
5053 Phone: 650-813-7696
5054 Fax: 650-813-6860
5055 e-mail: robert.herriot@pahv.xerox.com

5056
5057 Roger deBry
5058 [Utah Valley State College](#)
5059 [Orem, UT 84058](#)

5060
5061 [Phone: \(801\) 222-8000](#)
5062 [EMail: debryro@uvsc.edu](mailto:debryro@uvsc.edu)

5063 ~~[HUC/003G](#)~~
5064 ~~[IBM Corporation](#)~~
5065 ~~[P.O. Box 1900](#)~~
5066 ~~[Boulder, CO 80301-9191](#)~~

5067
5068 ~~[Phone: \(303\) 924-4080](#)~~
5069 ~~[Fax: \(303\) 924-9889](#)~~
5070 ~~[e-mail: debry@vnet.ibm.com](mailto:debry@vnet.ibm.com)~~

5071
5072 Patrick Powell
5073 Astart Technologies
5074 9475 Chesapeake Dr., Suite D
5075 San Diego, CA 95123

5076
5077 Phone: (619) 874-6543
5078 Fax: (619) 279-8424
5079 e-mail: papowell@astart.com

5080
5081 IPP Mailing List: ipp@pwg.org
5082 IPP Mailing List Subscription: ipp-request@pwg.org
5083 IPP Web Page: <http://www.pwg.org/ipp/>

5084
5085 Implementers of this specification are encouraged to join IPP Mailing List in order to participate in any
5086 discussions of clarification issues and review of registration proposals for additional attributes and
5087 values.

5088

5089 Other Participants:

5090 Chuck Adams - Tektronix
5091 Jeff Barnett - IBM
5092 Ron Bergman - Dataproducts Corp.
5093 Sylvan Butler - HP
5094 Keith Carter - IBM Corporation
5095 Jeff Copeland - QMS
5096 Andy Davidson - Tektronix
5097 Mabry Dozier - QMS
5098 Lee Farrell - Canon Information Systems
5099 Steve Gebert - IBM
5100 Babek Jahromi - Microsoft
5101 David Kellerman - Northlake Software
5102 Rick Landau - Digital
5103 Greg LeClair - Epson
5104 Harry Lewis - IBM
5105 Pete Loya - HP
5106 Ray Lutz - Cognisys
5107 Mike MacKay - Novell, Inc.
5108 Daniel Manchala - Xerox
5109 Carl-Uno Manros - Xerox
5110 Jay Martin - Underscore
5111 Larry Masinter - Xerox
5112 Stan McConnell - Xerox
5113 Ira McDonald - High North Inc.
5114 Paul Moore - Microsoft
5115 Tetsuya Morita - Ricoh
5116 Yuichi Niwa - Ricoh
5117 Pat Nogay - IBM
5118 Ron Norton - Printronics
5119 Bob Pentecost - HP
5120 Rob Rhoads - Intel
5121 Xavier Riley - Xerox
5122 David Roach - Unisys
5123 Stuart Rowley - Kyocera
5124 Hiroyuki Sato - Canon
5125 Bob Setterbo - Adobe
5126 Devon Taylor - Novell, Inc.
5127 Mike Timperman - Lexmark
5128 Randy Turner - Sharp
5129 Atsushi Yuki - Kyocera
5130 Rick Yardumian - Xerox
5131 Lloyd Young - Lexmark
5132 Bill Wagner - DPI

5133 Jim Walker - DAZEL
5134 Chris Wellens - Interworking Labs
5135 Rob Whittle - Novell, Inc.
5136 Don Wright - Lexmark
5137 Peter Zehler - Xerox
5138 Steve Zilles - Adobe

5139 11. Formats for IPP Registration Proposals

5140 In order to propose an IPP extension for registration, the proposer must submit an application to IANA
5141 by email to "iana@iana.org" or by filling out the appropriate form on the IANA web pages
5142 (<http://www.iana.org>). This section specifies the required information and the formats for proposing
5143 registrations of extensions to IPP as provided in Section 6 for:

- 5144
- 5145 1. type2 'keyword' attribute values
 - 5146 2. type3 'keyword' attribute values
 - 5147 3. type2 'enum' attribute values
 - 5148 4. type3 'enum' attribute values
 - 5149 5. attributes
 - 5150 6. attribute syntaxes
 - 5151 7. operations
 - 5152 8. status codes

5153 11.1 Type2 keyword attribute values registration

5154 Type of registration: type2 keyword attribute value
5155 Name of attribute to which this keyword specification is to be added:
5156 Proposed keyword name of this keyword value:
5157 Specification of this keyword value (follow the style of IPP Model Section 4.1.2.3):
5158 Name of proposer:
5159 Address of proposer:
5160 Email address of proposer:

5161
5162 Note: For type2 keywords, the Designated Expert will be the point of contact for the approved
5163 registration specification, if any maintenance of the registration specification is needed.

5164 11.2 Type3 keyword attribute values registration

5165 Type of registration: type3 keyword attribute value
5166 Name of attribute to which this keyword specification is to be added:
5167 Proposed keyword name of this keyword value:
5168 Specification of this keyword value (follow the style of IPP Model Section 4.1.2.3):
5169 Name of proposer:
5170 Address of proposer:
5171 Email address of proposer:

5172
5173 Note: For type3 keywords, the proposer will be the point of contact for the approved registration
5174 specification, if any maintenance of the registration specification is needed.

5175 11.3 Type2 enum attribute values registration

5176 Type of registration: type2 enum attribute value

5177 Name of attribute to which this enum specification is to be added:
5178 Keyword symbolic name of this enum value:
5179 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):
5180 Specification of this enum value (follow the style of IPP Model Section 4.1.4):
5181 Name of proposer:
5182 Address of proposer:
5183 Email address of proposer:
5184
5185 Note: For type2 enums, the Designated Expert will be the point of contact for the approved registration
5186 specification, if any maintenance of the registration specification is needed.

5187 11.4 Type3 enum attribute values registration

5188 Type of registration: type3 enum attribute value
5189 Name of attribute to which this enum specification is to be added:
5190 Keyword symbolic name of this enum value:
5191 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):
5192 Specification of this enum value (follow the style of IPP Model Section 4.1.4):
5193 Name of proposer:
5194 Address of proposer:
5195 Email address of proposer:
5196
5197 Note: For type3 enums, the proposer will be the point of contact for the approved registration
5198 specification, if any maintenance of the registration specification is needed.

5199 11.5 Attribute registration

5200 Type of registration: attribute
5201 Proposed keyword name of this attribute:
5202 Types of attribute (Operation, Job Template, Job Description, Printer Description):
5203 Operations to be used with if the attribute is an operation attribute:
5204 Object (Job, Printer, etc. if bound to an object):
5205 Attribute syntax(es) (include 1setOf and range as in Section 4.2):
5206 If attribute syntax is 'keyword' or 'enum', is it type2 or type3:
5207 If this is a Printer attribute, MAY the value returned depend on "document-format" (See Section 6.2):
5208 If this is a Job Template attribute, how does its specification depend on the value of the "multiple-
5209 document-handling" attribute:
5210 Specification of this attribute (follow the style of IPP Model Section 4.2):
5211 Name of proposer:
5212 Address of proposer:
5213 Email address of proposer:
5214
5215 Note: For attributes, the IPP Designated Expert will be the point of contact for the approved registration
5216 specification, if any maintenance of the registration specification is needed.

5217 11.6 Attribute Syntax registration

5218 Type of registration: attribute syntax

5219 Proposed name of this attribute syntax:

5220 Type of attribute syntax (integer, octetString, character-string, see [IPP-PRO]):

5221 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

5222 Specification of this attribute (follow the style of IPP Model Section 4.1):

5223 Name of proposer:

5224 Address of proposer:

5225 Email address of proposer:

5226

5227 Note: For attribute syntaxes, the IPP Designated Expert will be the point of contact for the approved

5228 registration specification, if any maintenance of the registration specification is needed.

5229 11.7 Operation registration

5230 Type of registration: operation

5231 Proposed name of this operation:

5232 Numeric operation-id value (to be assigned by the IPP Designated Expert in consultation with IANA):

5233 Object Target (Job, Printer, etc. that operation is upon):

5234 Specification of this attribute (follow the style of IPP Model Section 3):

5235 Name of proposer:

5236 Address of proposer:

5237 Email address of proposer:

5238

5239 Note: For operations, the IPP Designated Expert will be the point of contact for the approved

5240 registration specification, if any maintenance of the registration specification is needed.

5241 11.8 Attribute Group registration

5242 Type of registration: attribute group

5243 Proposed name of this attribute group:

5244 Numeric tag according to [IPP-PRO] (to be assigned by the IPP Designated Expert in consultation with
5245 IANA):

5246 Operation requests and group number for each operation in which the attribute group occurs:

5247 Operation responses and group number for each operation in which the attribute group occurs:

5248 Specification of this attribute group (follow the style of IPP Model Section 3):

5249 Name of proposer:

5250 Address of proposer:

5251 Email address of proposer:

5252

5253 Note: For attribute groups, the IPP Designated Expert will be the point of contact for the approved

5254 registration specification, if any maintenance of the registration specification is needed.

5255 11.9 Status code registration

5256 Type of registration: status code

5257 Keyword symbolic name of this status code value:

5258 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

5259 Operations that this status code may be used with:

5260 Specification of this status code (follow the style of IPP Model Section 13 APPENDIX B: Status Codes
5261 and Suggested Status Code Messages):

5262 Name of proposer:

5263 Address of proposer:

5264 Email address of proposer:

5265

5266 Note: For status codes, the Designated Expert will be the point of contact for the approved registration
5267 specification, if any maintenance of the registration specification is needed.

5268 12. APPENDIX A: Terminology

5269 This specification uses the terminology defined in this section.

5270 12.1 Conformance Terminology

5271 The key words "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT",
5272 "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in
5273 RFC 2119 [RFC2119].

5274 12.1.1 NEED NOT

5275 This term is not included in RFC 2119. The verb "NEED NOT" indicates an action that the subject of
5276 the sentence does not have to implement in order to claim conformance to the standard. The verb
5277 "NEED NOT" is used instead of "MAY NOT" since "MAY NOT" sounds like a prohibition.

5278 12.2 Model Terminology

5279 12.2.1 Keyword

5280 Keywords are used within this document as identifiers of semantic entities within the abstract model (see
5281 section 4.1.2.3). Attribute names, some attribute values, attribute syntaxes, and attribute group names
5282 are represented as keywords.

5283 12.2.2 Attributes

5284 An attribute is an item of information that is associated with an instance of an IPP object. An attribute
5285 consists of an attribute name and one or more attribute values. Each attribute has a specific attribute

5286 syntax. All object attributes are defined in section 3.3.5 and all operation attributes are defined in
5287 section 3.

5288 Job Template Attributes are described in section 4.2. The client optionally supplies Job Template
5289 attributes in a create request (operation requests that create Job objects). The Printer object has
5290 associated attributes which define supported and default values for the Printer.

5291 12.2.2.1 Attribute Name

5292 Each attribute is uniquely identified in this document by its attribute name. An attribute name is a
5293 keyword. The keyword attribute name is given in the section header describing that attribute. In running
5294 text in this document, attribute names are indicated inside double quotation marks (") where the
5295 quotation marks are not part of the keyword itself.

5296 12.2.2.2 Attribute Group Name

5297 Related attributes are grouped into named groups. The name of the group is a keyword. The group
5298 name may be used in place of naming all the attributes in the group explicitly. Attribute groups are
5299 defined in section 3.

5300 12.2.2.3 Attribute Value

5301 Each attribute has one or more values. Attribute values are represented in the syntax type specified for
5302 that attribute. In running text in this document, attribute values are indicated inside single quotation
5303 marks ('), whether their attribute syntax is keyword, integer, text, etc. where the quotation marks are not
5304 part of the value itself.

5305 12.2.2.4 Attribute Syntax

5306 Each attribute is defined using an explicit syntax type. In this document, each syntax type is defined as a
5307 keyword with specific meaning. The "Encoding and Transport" document [IPP-PRO] indicates the
5308 actual "on-the-wire" encoding rules for each syntax type. Attribute syntax types are defined in section
5309 4.1.

5310 12.2.3 Supports

5311 By definition, a Printer object supports an attribute only if that Printer object responds with the
5312 corresponding attribute populated with some value(s) in a response to a query for that attribute. A
5313 Printer object supports an attribute value if the value is one of the Printer object's "supported values"
5314 attributes. The device behind a Printer object may exhibit a behavior that corresponds to some IPP
5315 attribute, but if the Printer object, when queried for that attribute, doesn't respond with the attribute, then
5316 as far as IPP is concerned, that implementation does not support that feature. If the Printer object's "xxx-
5317 supported" attribute is not populated with a particular value (even if that value is a legal value for that
5318 attribute), then that Printer object does not support that particular value.

5319 A conforming implementation **MUST** support all **REQUIRED** attributes. However, even for
5320 **REQUIRED** attributes, conformance to IPP does not mandate that all implementations support all
5321 possible values representing all possible job processing behaviors and features. For example, if a given
5322 instance of a Printer supports only certain document formats, then that Printer responds with the
5323 "document-format-supported" attribute populated with a set of values, possibly only one, taken from the
5324 entire set of possible values defined for that attribute. This limited set of values represents the Printer's
5325 set of supported document formats. Supporting an attribute and some set of values for that attribute
5326 enables IPP end users to be aware of and make use of those features associated with that attribute and
5327 those values. If an implementation chooses to not support an attribute or some specific value, then IPP
5328 end users would have no ability to make use of that feature within the context of IPP itself. However,
5329 due to existing practice and legacy systems which are not IPP aware, there might be some other
5330 mechanism outside the scope of IPP to control or request the "unsupported" feature (such as embedded
5331 instructions within the document data itself).

5332 For example, consider the "finishings-supported" attribute.

- 5333 1) If a Printer object is not physically capable of stapling, the "finishings-supported" attribute **MUST**
5334 **NOT** be populated with the value of 'staple'.
- 5335 2) A Printer object is physically capable of stapling, however an implementation chooses not to
5336 support stapling in the IPP "finishings" attribute. In this case, 'staple' **MUST NOT** be a value in
5337 the "finishings-supported" Printer object attribute. Without support for the value 'staple', an IPP
5338 end user would have no means within the protocol itself to request that a Job be stapled.
5339 However, an existing document data formatter might be able to request that the document be
5340 stapled directly with an embedded instruction within the document data. In this case, the IPP
5341 implementation does not "support" stapling, however the end user is still able to have some
5342 control over the stapling of the completed job.
- 5343 3) A Printer object is physically capable of stapling, and an implementation chooses to support
5344 stapling in the IPP "finishings" attribute. In this case, 'staple' **MUST** be a value in the "finishings-
5345 supported" Printer object attribute. Doing so, would enable end users to be aware of and make
5346 use of the stapling feature using IPP attributes.
5347

5348 Even though support for Job Template attributes by a Printer object is **OPTIONAL**, it is
5349 **RECOMMENDED** that if the device behind a Printer object is capable of realizing any feature or
5350 function that corresponds to an IPP attribute and some associated value, then that implementation
5351 **SHOULD** support that IPP attribute and value.

5352 The set of values in any of the supported value attributes is set (populated) by some administrative
5353 process or automatic sensing mechanism that is outside the scope of this IPP/1.1 document. For
5354 administrative policy and control reasons, an administrator may choose to make only a subset of possible
5355 values visible to the end user. In this case, the real output device behind the IPP Printer abstraction may
5356 be capable of a certain feature, however an administrator is specifying that access to that feature not be
5357 exposed to the end user through the IPP protocol. Also, since a Printer object may represent a logical
5358 print device (not just a physical device) the actual process for supporting a value is undefined and left up
5359 to the implementation. However, if a Printer object supports a value, some manual human action may be
5360 needed to realize the semantic action associated with the value, but no end user action is required.

5361 For example, if one of the values in the "finishings-supported" attribute is 'staple', the actual process
5362 might be an automatic staple action by a physical device controlled by some command sent to the
5363 device. Or, the actual process of stapling might be a manual action by an operator at an operator
5364 attended Printer object.

5365 For another example of how supported attributes function, consider a system administrator who desires
5366 to control all print jobs so that no job sheets are printed in order to conserve paper. To force no job
5367 sheets, the system administrator sets the only supported value for the "job-sheets-supported" attribute to
5368 'none'. In this case, if a client requests anything except 'none', the create request is rejected or the "job-
5369 sheets" value is ignored (depending on the value of "ipp-attribute-fidelity"). To force the use of job
5370 start/end sheets on all jobs, the administrator does not include the value 'none' in the "job-sheets-
5371 supported" attribute. In this case, if a client requests 'none', the create request is rejected or the "job-
5372 sheets" value is ignored (again depending on the value of "ipp-attribute-fidelity").

5373 12.2.4 print-stream page

5374 A "print-stream page" is a page according to the definition of pages in the language used to express the
5375 document data.

5376 12.2.5 impression

5377 An "impression" is the image (possibly many print-stream pages in different configurations) imposed
5378 onto a single media page.

5379 13. APPENDIX B: Status Codes and Suggested Status Code Messages

5380 This section defines status code enum keywords and values that are used to provide semantic
5381 information on the results of an operation request. Each operation response **MUST** include a status
5382 code. The response **MAY** also contain a status message that provides a short textual description of the
5383 status. The status code is intended for use by automata, and the status message is intended for the human
5384 end user. Since the status message is an **OPTIONAL** component of the operation response, an IPP
5385 application (i.e., a browser, GUI, print driver or gateway) is **NOT REQUIRED** to examine or display the
5386 status message, since it **MAY** not be returned to the application.

5387 The prefix of the status keyword defines the class of response as follows:

5388 "informational" - Request received, continuing process
5389 "successful" - The action was successfully received, understood, and accepted
5390 "redirection" - Further action must be taken in order to complete the request
5391 "client-error" - The request contains bad syntax or cannot be fulfilled
5392 "server-error" - The IPP object failed to fulfill an apparently valid request
5393

5394 As with type2 enums, IPP status codes are extensible. IPP clients are **NOT REQUIRED** to understand
5395 the meaning of all registered status codes, though such understanding is obviously desirable. However,

5396 IPP clients MUST understand the class of any status code, as indicated by the prefix, and treat any
5397 unrecognized response as being equivalent to the first status code of that class, with the exception that an
5398 unrecognized response MUST NOT be cached. For example, if an unrecognized status code of "client-
5399 error-xxx-yyy" is received by the client, it can safely assume that there was something wrong with its
5400 request and treat the response as if it had received a "client-error-bad-request" status code. In such cases,
5401 IPP applications SHOULD present the OPTIONAL message (if present) to the end user since the
5402 message is likely to contain human readable information which will help to explain the unusual status.
5403 The name of the enum is the suggested status message for US English.

5404 The status code values range from 0x0000 to 0x7FFF. The value ranges for each status code class are as
5405 follows:

5406 "successful" - 0x0000 to 0x00FF
5407 "informational" - 0x0100 to 0x01FF
5408 "redirection" - 0x0200 to 0x02FF
5409 "client-error" - 0x0400 to 0x04FF
5410 "server-error" - 0x0500 to 0x05FF
5411

5412 The top half (128 values) of each range (0x0n40 to 0x0nFF, for n = 0 to 5) is reserved for private use
5413 within each status code class. Values 0x0600 to 0x7FFF are reserved for future assignment and MUST
5414 NOT be used.

5415 13.1 Status Codes

5416 Each status code is described below. Section 13.1.5.9 contains a table that indicates which status codes
5417 apply to which operations. The Implementer's Guide [IPP-IIG] describe the suggested steps for
5418 processing IPP attributes for all operations, including returning status codes.

5419 13.1.1 Informational

5420 This class of status code indicates a provisional response and is to be used for informational purposes
5421 only.

5422 There are no status codes defined in IPP/1.1 for this class of status code.

5423 13.1.2 Successful Status Codes

5424 This class of status code indicates that the client's request was successfully received, understood, and
5425 accepted.

5426 13.1.2.1 successful-ok (0x0000)

5427 The request has succeeded and no request attributes were substituted or ignored. In the case of a
5428 response to a create request, the 'successful-ok' status code indicates that the request was successfully
5429 received and validated, and that the Job object has been created; it does not indicate that the job has been

5430 processed. The transition of the Job object into the 'completed' state is the only indicator that the job has
5431 been printed.

5432 13.1.2.2 successful-ok-ignored-or-substituted-attributes (0x0001)

5433 The request has succeeded, but some supplied (1) attributes were ignored or (2) unsupported values were
5434 substituted with supported values or were ignored in order to perform the operation without rejecting it.
5435 Unsupported attributes, attribute syntaxes, or values MUST be returned in the Unsupported Attributes
5436 group of the response for all operations. There is an exception to this rule for the query operations: Get-
5437 Printer-Attributes, Get-Jobs, and Get-Job-Attributes for the "requested-attributes" operation attribute
5438 only. When the supplied values of the "requested-attributes" operation attribute are requesting attributes
5439 that are not supported, the IPP object MAY, but is NOT REQUIRED to, return the "requested-attributes"
5440 attribute in the Unsupported Attribute response group (with the unsupported values only). See sections [s](#)
5441 [3.1.7](#) [and](#) [3.2.1.2](#).

5442 13.1.2.3 successful-ok-conflicting-attributes (0x0002)

5443 The request has succeeded, but some supplied attribute values conflicted with the values of other
5444 supplied attributes. These conflicting values were either (1) substituted with (supported) values or (2)
5445 the attributes were removed in order to process the job without rejecting it. Attributes or values which
5446 conflict with other attributes and have been substituted or ignored MUST be returned in the Unsupported
5447 Attributes group of the response for all operations as supplied by the client. See sections [s](#) [3.1.7](#) [and](#)
5448 [3.2.1.2](#).

5449 13.1.3 Redirection Status Codes

5450 This class of status code indicates that further action needs to be taken to fulfill the request.

5451 There are no status codes defined in IPP/1.1 for this class of status code.

5452 13.1.4 Client Error Status Codes

5453 This class of status code is intended for cases in which the client seems to have erred. The IPP object
5454 SHOULD return a message containing an explanation of the error situation and whether it is a temporary
5455 or permanent condition.

5456 13.1.4.1 client-error-bad-request (0x0400)

5457 The request could not be understood by the IPP object due to malformed syntax (such as the value of a
5458 fixed length attribute whose length does not match the prescribed length for that attribute - see the
5459 Implementer's Guide [IPP-IIG]). The IPP application SHOULD NOT repeat the request without
5460 modifications.

5461 13.1.4.2 client-error-forbidden (0x0401)

5462 The IPP object understood the request, but is refusing to fulfill it. Additional authentication information
5463 or authorization credentials will not help and the request SHOULD NOT be repeated. This status code
5464 is commonly used when the IPP object does not wish to reveal exactly why the request has been refused
5465 or when no other response is applicable.

5466 13.1.4.3 client-error-not-authenticated (0x0402)

5467 The request requires user authentication. The IPP client may repeat the request with suitable
5468 authentication information. If the request already included authentication information, then this status
5469 code indicates that authorization has been refused for those credentials. If this response contains the
5470 same challenge as the prior response, and the user agent has already attempted authentication at least
5471 once, then the response message may contain relevant diagnostic information. This status codes reveals
5472 more information than "client-error-forbidden".

5473 13.1.4.4 client-error-not-authorized (0x0403)

5474 The requester is not authorized to perform the request. Additional authentication information or
5475 authorization credentials will not help and the request SHOULD NOT be repeated. This status code is
5476 used when the IPP object wishes to reveal that the authentication information is understandable,
5477 however, the requester is explicitly not authorized to perform the request. This status codes reveals
5478 more information than "client-error-forbidden" and "client-error-not-authenticated".

5479 13.1.4.5 client-error-not-possible (0x0404)

5480 This status code is used when the request is for something that can not happen. For example, there
5481 might be a request to cancel a job that has already been canceled or aborted by the system. The IPP
5482 client SHOULD NOT repeat the request.

5483 13.1.4.6 client-error-timeout (0x0405)

5484 The client did not produce a request within the time that the IPP object was prepared to wait. For
5485 example, a client issued a Create-Job operation and then, after a long period of time, issued a Send-
5486 Document operation and this error status code was returned in response to the Send-Document request
5487 (see section 3.3.1). The IPP object might have been forced to clean up resources that had been held for
5488 the waiting additional Documents. The IPP object was forced to close the Job since the client took too
5489 long. The client SHOULD NOT repeat the request without modifications.

5490 13.1.4.7 client-error-not-found (0x0406)

5491 The IPP object has not found anything matching the request URI. No indication is given of whether the
5492 condition is temporary or permanent. For example, a client with an old reference to a Job (a URI) tries
5493 to cancel the Job, however in the mean time the Job might have been completed and all record of it at the
5494 Printer has been deleted. This status code, 'client-error-not-found' is returned indicating that the

5495 referenced Job can not be found. This error status code is also used when a client supplies a URI as a
5496 reference to the document data in either a Print-URI or Send-URI operation, but the document can not be
5497 found.

5498 In practice, an IPP application should avoid a not found situation by first querying and presenting a list
5499 of valid Printer URIs and Job URIs to the end-user.

5500 13.1.4.8 client-error-gone (0x0407)

5501 The requested object is no longer available and no forwarding address is known. This condition should
5502 be considered permanent. Clients with link editing capabilities should delete references to the request
5503 URI after user approval. If the IPP object does not know or has no facility to determine, whether or not
5504 the condition is permanent, the status code "client-error-not-found" should be used instead.

5505 This response is primarily intended to assist the task of maintenance by notifying the recipient that the
5506 resource is intentionally unavailable and that the IPP object administrator desires that remote links to
5507 that resource be removed. It is not necessary to mark all permanently unavailable resources as "gone" or
5508 to keep the mark for any length of time -- that is left to the discretion of the IPP object administrator.

5509 13.1.4.9 client-error-request-entity-too-large (0x0408)

5510 The IPP object is refusing to process a request because the request entity is larger than the IPP object is
5511 willing or able to process. An IPP Printer returns this status code when it limits the size of print jobs and
5512 it receives a print job that exceeds that limit or when the attributes are so many that their encoding
5513 causes the request entity to exceed IPP object capacity.

5514 13.1.4.10 client-error-request-value-too-long (0x0409)

5515 The IPP object is refusing to service the request because one or more of the client-supplied attributes has
5516 a variable length value that is longer than the maximum length specified for that attribute. The IPP
5517 object might not have sufficient resources (memory, buffers, etc.) to process (even temporarily),
5518 interpret, and/or ignore a value larger than the maximum length. Another use of this error code is when
5519 the IPP object supports the processing of a large value that is less than the maximum length, but during
5520 the processing of the request as a whole, the object may pass the value onto some other system
5521 component which is not able to accept the large value. For more details, see the Implementer's Guide
5522 [IPP-IIG] .

5523 Note: For attribute values that are URIs, this rare condition is only likely to occur when a client has
5524 improperly submitted a request with long query information (e.g. an IPP application allows an end-user
5525 to enter an invalid URI), when the client has descended into a URI "black hole" of redirection (e.g., a
5526 redirected URI prefix that points to a suffix of itself), or when the IPP object is under attack by a client
5527 attempting to exploit security holes present in some IPP objects using fixed-length buffers for reading or
5528 manipulating the Request-URI.

5529 13.1.4.11 client-error-document-format-not-supported (0x040A)

5530 The IPP object is refusing to service the request because the document data is in a format, as specified in
5531 the "document-format" operation attribute, that is not supported by the Printer object. This error is
5532 returned independent of the client-supplied "ipp-attribute-fidelity". The Printer object MUST return this
5533 status code, even if there are other [Job Template](#) attributes that are not supported as well, since this error
5534 is a bigger problem than with Job Template attributes. [See sections 3.1.7 and 3.2.1.1. Issue 11. The
5535 Printer object MUST also return in the Unsupported Attributes Group the "document-format" attribute
5536 with the unsupported value supplied by the client. See section 3.2.1.](#)

5537 13.1.4.12 client-error-attributes-or-values-not-supported (0x040B)

5538 In a create request, if the Printer object does not support one or more attributes, attribute syntaxes, or
5539 attribute values supplied in the request and the client supplied the "ipp-attributes-fidelity" operation
5540 attribute with the 'true' value, the Printer object MUST return this status code. [The Printer object MUST
5541 also return in the Unsupported Attributes Group all the attributes and/or values supplied by the client
5542 that are not supported. See section 3.1.7. Issue 11](#) For example, if the request indicates 'iso-a4' media,
5543 but that media type is not supported by the Printer object. Or, if the client supplies [an optional a Job
5544 Template](#) attribute and the attribute itself is not even supported by the Printer. If the "ipp-attribute-
5545 fidelity" attribute is 'false', the Printer MUST ignore or substitute values for unsupported [Job Template](#)
5546 attributes and values rather than reject the request and return this status code.

5547 For any operation where a client requests attributes (such as a Get-Jobs, Get-Printer-Attributes, or Get-
5548 Job-Attributes operation), if the IPP object does not support one or more of the requested attributes, the
5549 IPP object simply ignores the unsupported requested attributes and processes the request as if they had
5550 not been supplied, rather than returning this status code. In this case, the IPP object MUST return the
5551 'successful-ok-ignored-or-substituted-attributes' status code and MAY return the unsupported attributes
5552 as values of the "requested-attributes" in the Unsupported Attributes Group (see section 13.1.2.2).

5553 13.1.4.13 client-error-uri-scheme-not-supported (0x040C)

5554 The [type-scheme](#) of the client-supplied URI in a Print-URI or a Send-URI operation is not supported.
5555 [See section 3.1.7. Issue 11](#)

5556 13.1.4.14 client-error-charset-not-supported (0x040D)

5557 For any operation, if the IPP Printer does not support the charset supplied by the client in the "attributes-
5558 charset" operation attribute, the Printer MUST reject the operation and return this status and any 'text' or
5559 'name' attributes using the 'utf-8' charset (see Section 3.1.4.1). [See section 3.1.7. Issue 11. The Printer
5560 object NEED NOT return other unsupported or conflicting attributes supplied by the client in the
5561 Unsupported Attributes Group. See section 3.2.1.2.](#)

5562 13.1.4.15 client-error-conflicting-attributes (0x040E)

5563 The request is rejected because some attribute values conflicted with the values of other attributes which
5564 this specification does not permit to be substituted or ignored. The Printer object MUST also return in
5565 the Unsupported Attributes Group the conflicting attributes supplied by the client. See sections 3.1.7
5566 and 3.2.1.2. Issue 27

5567 ~~14.1.4.16~~13.1.4.16 client-error-compression-not-supported (0x040F) Issue 6

5568 The IPP object is refusing to service the request because the document data, as specified in the
5569 "compression" operation attribute, is compressed in a way that is not supported by the Printer object.
5570 This error is returned independent of the client-supplied "ipp-attribute-fidelity". The Printer object
5571 MUST return this status code, even if there are other Job Template attributes that are not supported as
5572 well, since this error is a bigger problem than with Job Template attributes. Issue 6 See sections 3.1.7
5573 and 3.2.1.1. Issue 11~~The Printer object MUST also return in the Unsupported Attributes Group the~~
5574 ~~"compression" attribute with the unsupported value supplied by the client. See section 3.2.1. Issue 28~~

5575 ~~14.1.4.17~~13.1.4.17 client-error-compression-error (0x0410) Issue 6

5576 The IPP object is refusing to service the request because the document data cannot be decompressed
5577 when using the algorithm specified by the "compression" operation attribute. This error is returned
5578 independent of the client-supplied "ipp-attribute-fidelity". The Printer object MUST return this status
5579 code, even if there are other Job Template attributes that are not supported as well, since this error is a
5580 bigger problem than with Job Template attributes. See sections 3.1.7 and 3.2.1.1.

5581 ~~14.1.4.18~~13.1.4.18 client-error-document-format-error (0x0411) Issue 28

5582 The IPP object is refusing to service the request because Printer encountered an error in the document
5583 data while interpreting it. This error is returned independent of the client-supplied "ipp-attribute-
5584 fidelity". The Printer object MUST return this status code, even if there are other Job Template attributes
5585 that are not supported as well, since this error is a bigger problem than with Job Template attributes. See
5586 sections 3.1.7 and 3.2.1.1.

5587 13.1.4.19 client-error-document-access-error (0x0412) Issue 35

5588 The IPP object is refusing to service the Print-URI or Send-URI request because Printer encountered an
5589 access error while attempting to validate the accessibility or access the document data specified in the
5590 "document-uri" operation attribute. This error is returned independent of the client-supplied "ipp-
5591 attribute-fidelity". The Printer object MUST return this status code, even if there are Job Template
5592 attributes that are not supported as well, since this error is a bigger problem than with Job Template
5593 attributes. See section 3.1.7.

5594 13.1.5 Server Error Status Codes

5595 This class of status codes indicates cases in which the IPP object is aware that it has erred or is incapable
5596 of performing the request. The IPP object SHOULD include a message containing an explanation of the
5597 error situation, and whether it is a temporary or permanent condition.

5598 13.1.5.1 server-error-internal-error (0x0500)

5599 The IPP object encountered an unexpected condition that prevented it from fulfilling the request. This
5600 error status code differs from "server-error-temporary-error" in that it implies a more permanent type of
5601 internal error. It also differs from "server-error-device-error" in that it implies an unexpected condition
5602 (unlike a paper-jam or out-of-toner problem which is undesirable but expected). This error status code
5603 indicates that probably some knowledgeable human intervention is required.

5604 13.1.5.2 server-error-operation-not-supported (0x0501)

5605 The IPP object does not support the functionality required to fulfill the request. This is the appropriate
5606 response when the IPP object does not recognize an operation or is not capable of supporting it. [See](#)
5607 [section 3.1.7](#). [Issue 18](#)

5608 13.1.5.3 server-error-service-unavailable (0x0502)

5609 The IPP object is currently unable to handle the request due to a temporary overloading or maintenance
5610 of the IPP object. The implication is that this is a temporary condition which will be alleviated after
5611 some delay. If known, the length of the delay may be indicated in the message. If no delay is given, the
5612 IPP application should handle the response as it would for a "server-error-temporary-error" response. If
5613 the condition is more permanent, the error status codes "client-error-gone" or "client-error-not-found"
5614 could be used.

5615 13.1.5.4 server-error-version-not-supported (0x0503)

5616 The IPP object does not support, or refuses to support, the IPP protocol version that was [supplied as the](#)
5617 [value of the "version-number" operation parameter used](#) in the request ~~message~~. The IPP object is
5618 indicating that it is unable or unwilling to complete the request using the same [major and minor](#) version
5619 [number](#) as supplied in the request other than with this error message. The response ~~should~~ **SHOULD**
5620 contain a [Message "status-message" attribute](#) describing why that version is not supported and what
5621 other versions are supported by that IPP object. [See section 3.1.6](#).

5622 ~~A conforming IPP/1.1 client MUST specify a valid version ('1.1' or '1.0') on each request. A conforming~~
5623 ~~IPP/1.1 object MUST NOT return this status code to a conforming IPP/1.1 or IPP/1.0 client. An IPP~~
5624 ~~object MUST return this status code to a non-conforming IPP client.~~ The error response MUST identify
5625 in the "version-number" operation ~~attribute~~ [parameter](#) the closest version number that the IPP object
5626 does support. For example, if a client supplies version '1.0' ~~and, a conforming an~~ IPP/1.1 object [supports](#)
5627 [version '1.0', then it](#) MUST respond with version '1.0'. [If the IPP/1.1 object does not support version](#)
5628 ['1.0', then it MUST respond with this error code.](#) [Issue 36](#)

5629 13.1.5.5 server-error-device-error (0x0504)

5630 A printer error, such as a paper jam, occurs while the IPP object processes a Print or Send operation.
5631 The response contains the true Job Status (the values of the "job-state" and "job-state-reasons"
5632 attributes). Additional information can be returned in the OPTIONAL "job-state-message" attribute
5633 value or in the OPTIONAL status message that describes the error in more detail. This error status code
5634 is only returned in situations where the Printer is unable to accept the create request because of such a
5635 device error. For example, if the Printer is unable to spool, and can only accept one job at a time, the
5636 reason it might reject a create request is that the printer currently has a paper jam. In many cases
5637 however, where the Printer object can accept the request even though the Printer has some error
5638 condition, the 'successful-ok' status code will be returned. In such a case, the client would look at the
5639 returned Job Object Attributes or later query the Printer to determine its state and state reasons.

5640 13.1.5.6 server-error-temporary-error (0x0505)

5641 A temporary error such as a buffer full write error, a memory overflow (i.e. the document data exceeds
5642 the memory of the Printer), or a disk full condition, occurs while the IPP Printer processes an operation.
5643 The client MAY try the unmodified request again at some later point in time with an expectation that the
5644 temporary internal error condition may have been cleared. Alternatively, as an implementation option, a
5645 Printer object MAY delay the response until the temporary condition is cleared so that no error is
5646 returned.

5647 13.1.5.7 server-error-not-accepting-jobs (0x0506)

5648 A temporary error indicating that the Printer is not currently accepting jobs, because the administrator
5649 has set the value of the Printer's "printer-is-not-accepting-jobs" attribute to 'false' (by means outside the
5650 scope of this IPP/1.1 document).

5651 13.1.5.8 server-error-busy (0x0507)

5652 A temporary error indicating that the Printer is too busy processing jobs and/or other requests. The client
5653 SHOULD try the unmodified request again at some later point in time with an expectation that the
5654 temporary busy condition will have been cleared.

5655 13.1.5.9 server-error-job-canceled (0x0508)

5656 An error indicating that the job has been canceled by an operator or the system while the client was
5657 transmitting the data to the IPP Printer. If a job-id and job-uri had been created, then they are returned in
5658 the Print-Job, Send-Document, or Send-URI response as usual; otherwise, no job-id and job-uri are
5659 returned in the response.

5660 13.1.5.10 server-error-multiple-document-jobs-not-supported (0x0509) Issue 34

5661 The IPP object does not support multiple documents per job and a client attempted to supply document
5662 data with a second Send-Document or Send-URI operation.

5663 13.2 Status Codes for IPP Operations

5664 PJ = Print-Job, PU = Print-URI, CJ = Create-Job, SD = Send-Document
 5665 SU = Send-URI, V = Validate-Job, GA = Get-Job-Attributes and
 5666 Get-Printer-Attributes, GJ = Get-Jobs, C = Cancel-Job

5667 5668 5669	IPP Status Keyword	IPP Operations								
		PJ	PU	CJ	SD	SU	V	GA	GJ	C
5670	-----	--	--	--	--	--	--	--	--	--
5671	successful-ok	x	x	x	x	x	x	x	x	x
5672	successful-ok-ignored-or-substituted-	x	x	x	x	x	x	x	x	x
5673	attributes									
5674	successful-ok-conflicting-attributes	x	x	x	x	x	x	x	x	x
5675	client-error-bad-request	x	x	x	x	x	x	x	x	x
5676	client-error-forbidden	x	x	x	x	x	x	x	x	x
5677	client-error-not-authenticated	x	x	x	x	x	x	x	x	x
5678	client-error-not-authorized	x	x	x	x	x	x	x	x	x
5679	client-error-not-possible	x	x	x	x	x	x	x	x	x
5680	client-error-timeout				x	x				
5681	client-error-not-found	x	x	x	x	x	x	x	x	x
5682	client-error-gone	x	x	x	x	x	x	x	x	x
5683	client-error-request-entity-too-large	x	x	x	x	x	x	x	x	x
5684	client-error-request-value-too-long	x	x	x	x	x	x	x	x	x
5685	client-error-document-format-not-	x	x		x	x	x	x		
5686	supported									
5687	client-error-attributes-or-values-not-	x	x	x	x	x	x	x	x	x
5688	supported									
5689	client-error-uri-scheme-not-supported		x			x				
5690	client-error-charset-not-supported	x	x	x	x	x	x	x	x	x
5691	client-error-conflicting-attributes	x	x	x	x	x	x	x	x	x
5692	<u>client-error-compression-not-unsupported</u>		x	x		x	x	x		
5693	<u>client-error-compression-error</u>	x	x		x	x				
5694	<u>client-error-document-format-error</u>	x	x		x	x				
5695	<u>client-error-document-access-error</u>		x			x				
5696	server-error-internal-error	x	x	x	x	x	x	x	x	x
5697	server-error-operation-not-supported		x	x	x	x				
5698	server-error-service-unavailable	x	x	x	x	x	x	x	x	x
5699	server-error-version-not-supported	x	x	x	x	x	x	x	x	x
5700	server-error-device-error	x	x	x	x	x				
5701	server-error-temporary-error	x	x	x	x	x				
5702	server-error-not-accepting-jobs	x	x	x			x			
5703	server-error-busy	x	x	x	x	x	x	x	x	x
5704	server-error-job-canceled	x			x					
5705	<u>server-error-multiple-document-jobs-</u>				x	x				
5706	<u>not-supported</u>									

5707 HJ = Hold-Job, RJ = Release-Job, RS = Restart-Job
 5708 PP = Pause-Printer, RP = Resume-Printer, PJ = Purge-Jobs

5709		IPP Operations (cont.)					
5710	IPP Status Keyword	HJ	RJ	RS	PP	RP	PJ
5711	-----	--	--	--	--	--	--
5712	successful-ok	x	x	x	x	x	x
5713	successful-ok-ignored-or-substituted-	x	x	x	x	x	x
5714	attributes						
5715	successful-ok-conflicting-attributes	x	x	x	x	x	x
5716	client-error-bad-request	x	x	x	x	x	x
5717	client-error-forbidden	x	x	x	x	x	x
5718	client-error-not-authenticated	x	x	x	x	x	x
5719	client-error-not-authorized	x	x	x	x	x	x
5720	client-error-not-possible	x	x	x	x	x	x
5721	client-error-timeout						
5722	client-error-not-found	x	x	x	x	x	x
5723	client-error-gone	x	x	x	x	x	x
5724	client-error-request-entity-too-large	x	x	x	x	x	x
5725	client-error-request-value-too-long	x	x	x	x	x	x
5726	client-error-document-format-not-						
5727	supported						
5728	client-error-attributes-or-values-not-	x	x	x	x	x	x
5729	supported						
5730	client-error-uri-scheme-not-supported						
5731	client-error-charset-not-supported	x	x	x	x	x	x
5732	client-error-conflicting-attributes	x	x	x	x	x	x
5733	<u>client-error-compression-not-supported</u>						
5734	<u>client-error-compression-error</u>						
5735	<u>client-error-document-format-error</u>						
5736	<u>client-error-document-access-error</u>						
5737	server-error-internal-error	x	x	x	x	x	x
5738	server-error-operation-not-supported	x	x	x	x	x	x
5739	server-error-service-unavailable	x	x	x	x	x	x
5740	server-error-version-not-supported	x	x	x	x	x	x
5741	server-error-device-error						
5742	server-error-temporary-error						
5743	server-error-not-accepting-jobs						
5744	server-error-busy	x	x	x	x	x	x
5745	server-error-job-canceled						
5746	<u>server-error-multiple-document-jobs-</u>						
5747	<u>not-supported</u>						
5748							
5749							

5750

5751 14. APPENDIX C: "media" keyword values

5752 Standard keyword values are taken from several sources.

5753 Standard values are defined (taken from DPA[ISO10175] and the Printer MIB[RFC1759]):

5754 `default`: The default medium for the output device
5755 `iso-a4-white`: Specifies the ISO A4 white medium
5756 `iso-a4-colored`: Specifies the ISO A4 colored medium
5757 `iso-a4-transparent`: Specifies the ISO A4 transparent medium
5758 `iso-a3-white`: Specifies the ISO A3 white medium
5759 `iso-a3-colored`: Specifies the ISO A3 colored medium
5760 `iso-a5-white`: Specifies the ISO A5 white medium
5761 `iso-a5-colored`: Specifies the ISO A5 colored medium
5762 `iso-b4-white`: Specifies the ISO B4 white medium
5763 `iso-b4-colored`: Specifies the ISO B4 colored medium
5764 `iso-b5-white`: Specifies the ISO B5 white medium
5765 `iso-b5-colored`: Specifies the ISO B5 colored medium
5766 `jis-b4-white`: Specifies the JIS B4 white medium
5767 `jis-b4-colored`: Specifies the JIS B4 colored medium
5768 `jis-b5-white`: Specifies the JIS B5 white medium
5769 `jis-b5-colored`: Specifies the JIS B5 colored medium

5770

5771 The following standard values are defined for North American media:

5772 `na-letter-white`: Specifies the North American letter white medium
5773 `na-letter-colored`: Specifies the North American letter colored medium
5774 `na-letter-transparent`: Specifies the North American letter transparent medium
5775 `na-legal-white`: Specifies the North American legal white medium
5776 `na-legal-colored`: Specifies the North American legal colored medium

5777

5778 The following standard values are defined for envelopes:

5779 `iso-b4-envelope`: Specifies the ISO B4 envelope medium
5780 `iso-b5-envelope`: Specifies the ISO B5 envelope medium
5781 `iso-c3-envelope`: Specifies the ISO C3 envelope medium
5782 `iso-c4-envelope`: Specifies the ISO C4 envelope medium
5783 `iso-c5-envelope`: Specifies the ISO C5 envelope medium
5784 `iso-c6-envelope`: Specifies the ISO C6 envelope medium
5785 `iso-designated-long-envelope`: Specifies the ISO Designated Long envelope medium
5786 `na-10x13-envelope`: Specifies the North American 10x13 envelope medium
5787 `na-9x12-envelope`: Specifies the North American 9x12 envelope medium

5788 'monarch-envelope': Specifies the Monarch envelope
5789 'na-number-10-envelope': Specifies the North American number 10 business envelope medium
5790 'na-7x9-envelope': Specifies the North American 7x9 inch envelope
5791 'na-9x11-envelope': Specifies the North American 9x11 inch envelope
5792 'na-10x14-envelope': Specifies the North American 10x14 inch envelope
5793 'na-number-9-envelope': Specifies the North American number 9 business envelope
5794 'na-6x9-envelope': Specifies the North American 6x9 inch envelope
5795 'na-10x15-envelope': Specifies the North American 10x15 inch envelope
5796

5797 The following standard values are defined for the less commonly used media (white-only):

5798 'executive-white': Specifies the white executive medium
5799 'folio-white': Specifies the folio white medium
5800 'invoice-white': Specifies the white invoice medium
5801 'ledger-white': Specifies the white ledger medium
5802 'quarto-white': Specifies the white quarto medium
5803 'iso-a0-white': Specifies the ISO A0 white medium
5804 'iso-a1-white': Specifies the ISO A1 white medium
5805 'iso-a2-white': Specifies the ISO A2 white medium
5806 'iso-a6-white': Specifies the ISO A6 white medium
5807 'iso-a7-white': Specifies the ISO A7 white medium
5808 'iso-a8-white': Specifies the ISO A8 white medium
5809 'iso-a9-white': Specifies the ISO A9 white medium
5810 'iso-10-white': Specifies the ISO A10 white medium
5811 'iso-b0-white': Specifies the ISO B0 white medium
5812 'iso-b1-white': Specifies the ISO B1 white medium
5813 'iso-b2-white': Specifies the ISO B2 white medium
5814 'iso-b3-white': Specifies the ISO B3 white medium
5815 'iso-b6-white': Specifies the ISO B6 white medium
5816 'iso-b7-white': Specifies the ISO B7 white medium
5817 'iso-b8-white': Specifies the ISO B8 white medium
5818 'iso-b9-white': Specifies the ISO B9 white medium
5819 'iso-b10-white': Specifies the ISO B10 white medium
5820 'jis-b0-white': Specifies the JIS B0 white medium
5821 'jis-b1-white': Specifies the JIS B1 white medium
5822 'jis-b2-white': Specifies the JIS B2 white medium
5823 'jis-b3-white': Specifies the JIS B3 white medium
5824 'jis-b6-white': Specifies the JIS B6 white medium
5825 'jis-b7-white': Specifies the JIS B7 white medium
5826 'jis-b8-white': Specifies the JIS B8 white medium
5827 'jis-b9-white': Specifies the JIS B9 white medium
5828 'jis-b10-white': Specifies the JIS B10 white medium
5829

5830 The following standard values are defined for engineering media:

5831 `a`: Specifies the engineering A size medium
5832 `b`: Specifies the engineering B size medium
5833 `c`: Specifies the engineering C size medium
5834 `d`: Specifies the engineering D size medium
5835 `e`: Specifies the engineering E size medium
5836

5837 The following standard values are defined for input-trays (from ISO DPA and the Printer MIB):

5838 `top`: The top input tray in the printer.
5839 `middle`: The middle input tray in the printer.
5840 `bottom`: The bottom input tray in the printer.
5841 `envelope`: The envelope input tray in the printer.
5842 `manual`: The manual feed input tray in the printer.
5843 `large-capacity`: The large capacity input tray in the printer.
5844 `main`: The main input tray
5845 `side`: The side input tray
5846

5847 The following standard values are defined for media sizes (from ISO DPA):

5848 `iso-a0`: Specifies the ISO A0 size: 841 mm by 1189 mm as defined in ISO 216
5849 `iso-a1`: Specifies the ISO A1 size: 594 mm by 841 mm as defined in ISO 216
5850 `iso-a2`: Specifies the ISO A2 size: 420 mm by 594 mm as defined in ISO 216
5851 `iso-a3`: Specifies the ISO A3 size: 297 mm by 420 mm as defined in ISO 216
5852 `iso-a4`: Specifies the ISO A4 size: 210 mm by 297 mm as defined in ISO 216
5853 `iso-a5`: Specifies the ISO A5 size: 148 mm by 210 mm as defined in ISO 216
5854 `iso-a6`: Specifies the ISO A6 size: 105 mm by 148 mm as defined in ISO 216
5855 `iso-a7`: Specifies the ISO A7 size: 74 mm by 105 mm as defined in ISO 216
5856 `iso-a8`: Specifies the ISO A8 size: 52 mm by 74 mm as defined in ISO 216
5857 `iso-a9`: Specifies the ISO A9 size: 37 mm by 52 mm as defined in ISO 216
5858 `iso-a10`: Specifies the ISO A10 size: 26 mm by 37 mm as defined in ISO 216
5859 `iso-b0`: Specifies the ISO B0 size: 1000 mm by 1414 mm as defined in ISO 216
5860 `iso-b1`: Specifies the ISO B1 size: 707 mm by 1000 mm as defined in ISO 216
5861 `iso-b2`: Specifies the ISO B2 size: 500 mm by 707 mm as defined in ISO 216
5862 `iso-b3`: Specifies the ISO B3 size: 353 mm by 500 mm as defined in ISO 216
5863 `iso-b4`: Specifies the ISO B4 size: 250 mm by 353 mm as defined in ISO 216
5864 `iso-b5`: Specifies the ISO B5 size: 176 mm by 250 mm as defined in ISO 216
5865 `iso-b6`: Specifies the ISO B6 size: 125 mm by 176 mm as defined in ISO 216
5866 `iso-b7`: Specifies the ISO B7 size: 88 mm by 125 mm as defined in ISO 216
5867 `iso-b8`: Specifies the ISO B8 size: 62 mm by 88 mm as defined in ISO 216
5868 `iso-b9`: Specifies the ISO B9 size: 44 mm by 62 mm as defined in ISO 216
5869 `iso-b10`: Specifies the ISO B10 size: 31 mm by 44 mm as defined in ISO 216
5870 `na-letter`: Specifies the North American letter size: 8.5 inches by 11 inches
5871 `na-legal`: Specifies the North American legal size: 8.5 inches by 14 inches
5872 `executive`: Specifies the executive size (7.25 X 10.5 in)
5873 `folio`: Specifies the folio size (8.5 X 13 in)

5874 'invoice': Specifies the invoice size (5.5 X 8.5 in)
5875 'ledger': Specifies the ledger size (11 X 17 in)
5876 'quarto': Specifies the quarto size (8.5 X 10.83 in)
5877 'iso-c3': Specifies the ISO C3 size: 324 mm by 458 mm as defined in ISO 269
5878 'iso-c4': Specifies the ISO C4 size: 229 mm by 324 mm as defined in ISO 269
5879 'iso-c5': Specifies the ISO C5 size: 162 mm by 229 mm as defined in ISO 269
5880 'iso-c6': Specifies the ISO C6 size: 114 mm by 162 mm as defined in ISO 269
5881 'iso-designated-long': Specifies the ISO Designated Long size: 110 mm by 220 mm as defined in ISO
5882 269
5883 'na-10x13-envelope': Specifies the North American 10x13 size: 10 inches by 13 inches
5884 'na-9x12-envelope': Specifies the North American 9x12 size: 9 inches by 12 inches
5885 'na-number-10-envelope': Specifies the North American number 10 business envelope size: 4.125
5886 inches by 9.5 inches
5887 'na-7x9-envelope': Specifies the North American 7x9 inch envelope size
5888 'na-9x11-envelope': Specifies the North American 9x11 inch envelope size
5889 'na-10x14-envelope': Specifies the North American 10x14 inch envelope size
5890 'na-number-9-envelope': Specifies the North American number 9 business envelope size
5891 'na-6x9-envelope': Specifies the North American 6x9 envelope size
5892 'na-10x15-envelope': Specifies the North American 10x15 envelope size
5893 'monarch-envelope': Specifies the Monarch envelope size (3.87 x 7.5 in)
5894 'jis-b0': Specifies the JIS B0 size: 1030mm x 1456mm
5895 'jis-b1': Specifies the JIS B1 size: 728mm x 1030mm
5896 'jis-b2': Specifies the JIS B2 size: 515mm x 728mm
5897 'jis-b3': Specifies the JIS B3 size: 364mm x 515mm
5898 'jis-b4': Specifies the JIS B4 size: 257mm x 364mm
5899 'jis-b5': Specifies the JIS B5 size: 182mm x 257mm
5900 'jis-b6': Specifies the JIS B6 size: 128mm x 182mm
5901 'jis-b7': Specifies the JIS B7 size: 91mm x 128mm
5902 'jis-b8': Specifies the JIS B8 size: 64mm x 91mm
5903 'jis-b9': Specifies the JIS B9 size: 45mm x 64mm
5904 'jis-b10': Specifies the JIS B10 size: 32mm x 45mm

5905 15. APPENDIX D: Processing IPP Attributes

5906 When submitting a print job to a Printer object, the IPP model allows a client to supply operation and
5907 Job Template attributes along with the document data. These Job Template attributes in the create
5908 request affect the rendering, production and finishing of the documents in the job. Similar types of
5909 instructions may also be contained in the document to be printed, that is, embedded within the print data
5910 itself. In addition, the Printer has a set of attributes that describe what rendering and finishing options
5911 which are supported by that Printer. This model, which allows for flexibility and power, also introduces
5912 the potential that at job submission time, these client-supplied attributes may conflict with either:

- 5913 - what the implementation is capable of realizing (i.e., what the Printer supports), as well as
- 5914 - the instructions embedded within the print data itself.

5915

5916 The following sections describe how these two types of conflicts are handled in the IPP model.

5917 15.1 Fidelity

5918 If there is a conflict between what the client requests and what a Printer object supports, the client may
5919 request one of two possible conflict handling mechanisms:

- 5920 1) either reject the job since the job can not be processed exactly as specified, or
 - 5921 2) allow the Printer to make any changes necessary to proceed with processing the Job the best it can.
- 5922

5923 In the first case the client is indicating to the Printer object: "Print the job exactly as specified with no
5924 exceptions, and if that can't be done, don't even bother printing the job at all." In the second case, the
5925 client is indicating to the Printer object: "It is more important to make sure the job is printed rather than
5926 be processed exactly as specified; just make sure the job is printed even if client supplied attributes need
5927 to be changed or ignored."

5928 The IPP model accounts for this situation by introducing an "ipp-attribute-fidelity" attribute.

5929 In a create request, "ipp-attribute-fidelity" is a boolean operation attribute that is **OPTIONALLY**
5930 supplied by the client. The value 'true' indicates that total fidelity to client supplied Job Template
5931 attributes and values is required. The client is requesting that the Job be printed exactly as specified, and
5932 if that is not possible then the job **MUST** be rejected rather than processed incorrectly. The value 'false'
5933 indicates that a reasonable attempt to print the Job is acceptable. If a Printer does not support some of
5934 the client supplied Job Template attributes or values, the Printer **MUST** ignore them or substitute any
5935 supported value for unsupported values, respectively. The Printer may choose to substitute the default
5936 value associated with that attribute, or use some other supported value that is similar to the unsupported
5937 requested value. For example, if a client supplies a "media" value of 'na-letter', the Printer may choose
5938 to substitute 'iso-a4' rather than a default value of 'envelope'. If the client does not supply the "ipp-
5939 attribute-fidelity" attribute, the Printer assumes a value of 'false'.

5940 Each Printer implementation **MUST** support both types of "fidelity" printing (that is whether the client
5941 supplies a value of 'true' or 'false'):

- 5942 - If the client supplies 'false' or does not supply the attribute, the Printer object **MUST** always accept
5943 the request by ignoring unsupported Job Template attributes and by substituting unsupported
5944 values of supported Job Template attributes with supported values.
 - 5945 - If the client supplies 'true', the Printer object **MUST** reject the request if the client supplies
5946 unsupported Job Template attributes.
- 5947

5948 Since a client can always query a Printer to find out exactly what is and is not supported, "ipp-attribute-
5949 fidelity" set to 'false' is useful when:

- 5950 1) The End-User uses a command line interface to request attributes that might not be supported.
- 5951 2) In a GUI context, if the End User expects the job might be moved to another printer and prefers a
5952 sub-optimal result to nothing at all.
- 5953 3) The End User just wants something reasonable in lieu of nothing at all.

5954

5955 15.2 Page Description Language (PDL) Override

5956 If there is a conflict between the value of an IPP Job Template attribute and a corresponding instruction
5957 in the document data, the value of the IPP attribute SHOULD take precedence over the document
5958 instruction. Consider the case where a previously formatted file of document data is sent to an IPP
5959 Printer. In this case, if the client supplies any attributes at job submission time, the client desires that
5960 those attributes override the embedded instructions. Consider the case were a previously formatted
5961 document has embedded in it commands to load 'iso-a4' media. However, the document is passed to an
5962 end user that only has access to a printer with 'na-letter' media loaded. That end user most likely wants
5963 to submit that document to an IPP Printer with the "media" Job Template attribute set to 'na-letter'. The
5964 job submission attribute should take precedence over the embedded PDL instruction. However, until
5965 companies that supply document data interpreters allow a way for external IPP attributes to take
5966 precedence over embedded job production instructions, a Printer might not be able to support the
5967 semantics that IPP attributes override the embedded instructions.

5968 The IPP model accounts for this situation by introducing a "pdl-override-supported" attribute that
5969 describes the Printer objects capabilities to override instructions embedded in the PDL data stream. The
5970 value of the "pdl-override-supported" attribute is configured by means outside the scope of this IPP/1.1
5971 document.

5972 This REQUIRED Printer attribute takes on the following values:

- 5973 - 'attempted': This value indicates that the Printer object attempts to make the IPP attribute values
5974 take precedence over embedded instructions in the document data, however there is no guarantee.
- 5975 - 'not-attempted': This value indicates that the Printer object makes no attempt to make the IPP
5976 attribute values take precedence over embedded instructions in the document data.
5977

5978 At job processing time, an implementation that supports the value of 'attempted' might do one of several
5979 different actions:

- 5980 1) Generate an output device specific command sequence to realize the feature represented by the
5981 IPP attribute value.
- 5982 2) Parse the document data itself and replace the conflicting embedded instruction with a new
5983 embedded instruction that matches the intent of the IPP attribute value.
- 5984 3) Indicate to the Printer that external supplied attributes take precedence over embedded instructions
5985 and then pass the external IPP attribute values to the document data interpreter.
- 5986 4) Anything else that allows for the semantics that IPP attributes override embedded document data
5987 instructions.
5988

5989 Since 'attempted' does not offer any type of guarantee, even though a given Printer object might not do a
5990 very "good" job of attempting to ensure that IPP attributes take a higher precedence over instructions
5991 embedded in the document data, it would still be a conforming implementation.

5992 At job processing time, an implementation that supports the value of 'not-attempted' might do one of the
5993 following actions:

- 5994 1) Simply pre-pend the document data with the PDL instruction that corresponds to the client-
5995 supplied PDL attribute, such that if the document data also has the same PDL instruction, it will
5996 override what the Printer object pre-pended. In other words, this implementation is using the
5997 same implementation semantics for the client-supplied IPP attributes as for the Printer object
5998 defaults.
- 5999 2) Parse the document data and replace the conflicting embedded instruction with a new embedded
6000 instruction that approximates, but does not match, the semantic intent of the IPP attribute value.
6001

6002 Note: The "ipp-attribute-fidelity" attribute applies to the Printer's ability to either accept or reject other
6003 unsupported Job Template attributes. In other words, if "ipp-attribute-fidelity" is set to 'true', a Job is
6004 accepted if and only if the client supplied Job Template attributes and values are supported by the
6005 Printer. Whether these attributes actually affect the processing of the Job when the document data
6006 contains embedded instructions depends on the ability of the Printer to override the instructions
6007 embedded in the document data with the semantics of the IPP attributes. If the document data attributes
6008 can be overridden ("pdl-override-supported" set to 'attempted'), the Printer makes an attempt to use the
6009 IPP attributes when processing the Job. If the document data attributes can not be overridden ("pdl-
6010 override-supported" set to 'not-attempted'), the Printer makes no attempt to override the embedded
6011 document data instructions with the IPP attributes when processing the Job, and hence, the IPP attributes
6012 may fail to affect the Job processing and output when the corresponding instruction is embedded in the
6013 document data.

6014 15.3 Using Job Template Attributes During Document Processing.

6015 The Printer object uses some of the Job object's Job Template attributes during the processing of the
6016 document data associated with that job. These include, but are not limited to, "orientation-requested",
6017 "number-up", "sides", "media", and "copies". The processing of each document in a Job Object MUST
6018 follow the steps below. These steps are intended only to identify when and how attributes are to be used
6019 in processing document data and any alternative steps that accomplishes the same effect can be used to
6020 implement this specification.

- 6021 1. Using the client supplied "document-format" attribute or some form of document format detection
6022 algorithm (if the value of "document-format" is not specific enough), determine whether or not
6023 the document data has already been formatted for printing. If the document data has been
6024 formatted, then go to step 2. Otherwise, the document data MUST be formatted. The formatting
6025 detection algorithm is implementation defined and is not specified by this specification. The
6026 formatting of the document data uses the "orientation-requested" attribute to determine how the
6027 formatted print data should be placed on a print-stream page, see section 4.2.10 for the details.
6028
- 6029 2. The document data is in the form of a print-stream in a known media type. The "page-ranges"
6030 attribute is used to select, as specified in section 4.2.7, a sub-sequence of the pages in the print-
6031 stream that are to be processed and images.
6032

6033 3. The input to this step is a sequence of print-stream pages. This step is controlled by the "number-
6034 up" attribute. If the value of "number-up" is N, then during the processing of the print-stream
6035 pages, each N print-stream pages are positioned, as specified in section 4.2.9, to create a single
6036 impression. If a given document does not have N more print-stream pages, then the completion
6037 of the impression is controlled by the "multiple-document-handling" attribute as described in
6038 section 4.2.4; when the value of this attribute is 'single-document' or 'single-document-new-
6039 sheet', the print-stream pages of document data from subsequent documents is used to complete
6040 the impression.

6041
6042 The size(scaling), position(translation) and rotation of the print-stream pages on the impression is
6043 implementation defined. Note that during this process the print-stream pages may be rendered to
6044 a form suitable for placing on the impression; this rendering is controlled by the values of the
6045 "printer-resolution" and "print-quality" attributes as described in sections 4.2.12 and 4.2.13. In
6046 the case N=1, the impression is nearly the same as the print-stream page; the differences would
6047 only be in the size, position and rotation of the print-stream page and/or any decoration, such as a
6048 frame to the page, that is added by the implementation.

6049
6050 4. The collection of impressions is placed, in sequence, onto sides of the media sheets. This
6051 placement is controlled by the "sides" attribute and the orientation of the print-stream page, as
6052 described in section 4.2.8. The orientation of the print-stream pages affects the orientation of the
6053 impression; for example, if "number-up" equals 2, then, typically, two portrait print-stream pages
6054 become one landscape impression. Note that the placement of impressions onto media sheets is
6055 also controlled by the "multiple-document-handling" attribute as described in section 4.2.4.

6056
6057 5. The "copies" and "multiple-document-handling" attributes are used to determine how many copies
6058 of each media instance are created and in what order. See sections 4.2.5 and 4.2.4 for the details.

6059
6060 6. When the correct number of copies are created, the media instances are finished according to the
6061 values of the "finishings" attribute as described in 4.2.6. Note that sometimes finishing
6062 operations may require manual intervention to perform the finishing operations on the copies,
6063 especially uncollated copies. This specification allows any or all of the processing steps to be
6064 performed automatically or manually at the discretion of the Printer object.

6065 16. APPENDIX E: Generic Directory Schema

6066 This section defines a generic schema for an entry in a directory service. A directory service is a means
6067 by which service users can locate service providers. In IPP environments, this means that IPP Printers
6068 can be registered (either automatically or with the help of an administrator) as entries of type printer in
6069 the directory using an implementation specific mechanism such as entry attributes, entry type fields,
6070 specific branches, etc. IPP clients can search or browse for entries of type printer. Clients use the
6071 directory service to find entries based on naming, organizational contexts, or filtered searches on
6072 attribute values of entries. For example, a client can find all printers in the "Local Department" context.
6073 Authentication and authorization are also often part of a directory service so that an administrator can

6074 place limits on end users so that they are only allowed to find entries to which they have certain access
6075 rights. IPP itself does not require any specific directory service protocol or provider.

6076 Note: Some directory implementations allow for the notion of "aliasing". That is, one directory entry
6077 object can appear as multiple directory entry object with different names for each object. In each case,
6078 each alias refers to the same directory entry object which refers to a single IPP Printer object.

6079 The generic schema is a subset of IPP Printer Job Template and Printer Description attributes (sections
6080 4.2 and 4.4). These attributes are identified as either RECOMMENDED or OPTIONAL for the
6081 directory entry itself. This conformance labeling is NOT the same conformance labeling applied to the
6082 attributes of IPP Printers objects. The conformance labeling in this Appendix is intended to apply to
6083 directory templates and to IPP Printer implementations that subscribe by adding one or more entries to a
6084 directory. RECOMMENDED attributes SHOULD be associated with each directory entry. OPTIONAL
6085 attributes MAY be associated with the directory entry (if known or supported). In addition, all directory
6086 entry attributes SHOULD reflect the current attribute values for the corresponding Printer object.

6087 The names of attributes in directory schema and entries SHOULD be the same as the IPP Printer
6088 attribute names as shown.

6089 In order to bridge between the directory service and the IPP Printer object, one of the RECOMMENDED
6090 directory entry attributes is the Printer object's "printer-uri-supported" attribute. The IPP client queries
6091 the "printer-uri-supported" attribute in the directory entry and then addresses the IPP Printer object using
6092 one of its URIs. The "uri-security-supported" attribute identifies the protocol (if any) used to secure a
6093 channel.

6094 The following attributes define the generic schema for directory entries of type PRINTER:

6095	printer-uri-supported	RECOMMENDED	Section 4.4.1
6096	<u>uri-authentication-supported</u>	<u>RECOMMENDED</u>	<u>Section 4.4.2</u>
6097	uri-security-supported	RECOMMENDED	Section 4.4.3 4.4.2
6098	printer-name	RECOMMENDED	Section 4.4.4
6099	printer-location	RECOMMENDED	Section 4.4.5
6100	printer-info	OPTIONAL	Section 4.4.6
6101	printer-more-info	OPTIONAL	Section 4.4.7
6102	printer-make-and-model	RECOMMENDED	Section 4.4.9
6103	charset-supported	OPTIONAL	Section 4.4.18
6104	generated-natural-language-		
6105	supported	OPTIONAL	Section 4.4.20
6106	document-format-supported	RECOMMENDED	Section 4.4.22
6107	<u>compression-supported</u>	<u>RECOMMENDED</u>	<u>Section 4.4.32</u>
6108	color-supported	RECOMMENDED	Section 4.4.26
6109	finishings-supported	OPTIONAL	Section 4.2.6
6110	number-up-supported	OPTIONAL	Section 4.2.7
6111	sides-supported	RECOMMENDED	Section 4.2.8
6112	media-supported	RECOMMENDED	Section 4.2.11
6113	printer-resolution-supported	OPTIONAL	Section 4.2.12
6114	print-quality-supported	OPTIONAL	Section 4.2.13

6115	<u>ipp-versions-supported</u>	RECOMMENDED	<u>Section</u> 4.4.14
6116	<u>multiple-document-jobs-supported</u>	OPTIONAL	<u>Section</u> 4.4.16
6117	pages-per-minute	OPTIONAL	Section 4.4.36
6118	pages-per-minute-color	OPTIONAL	Section 4.4.37

6119

6120 17. APPENDIX F: Differences between the IPP/1.0 and IPP/1.1 "Model and Semantics" Specifications

6121 This Appendix is divided into two lists that summarize the differences between IPP/1.1 (this document)
 6122 and IPP/1.0 [RFC2566]. The section numbers refer to the numbers in this document which in some
 6123 cases have changed from RFC 2566. When a change affects multiple sections, the item is listed once in
 6124 the order of the first section affected and the remaining affected section numbers are indicated.

6125 The first list contains extensions and clarifications and the second list contains changes in semantics or
 6126 conformance. However, note that client and IPP object implementations of IPP/1.0 MAY implement
 6127 any of the extensions and clarifications in this document.

6128 The following ~~IPP/1.0 [IPP-MOD1.0]~~ extensions and clarifications have been incorporated into
 6129 ~~IPP/1.1~~ this document:

- 6130 1. Section 2.1 - clarified that the term "client" can be either contained in software controlled by an
 6131 end user or a part of a print server that controls devices. Issue 4 and Issue 5
- 6132 2. Section 2.4 - added the description of the new "uri-authentication-supported" Printer Description
 6133 attribute. Issue 2
- 6134 3. Section 3.1.3, 3.1.6, 3.2.5.2, and - clarified the error handling for operation attributes that have
 6135 their own status code. Issues 18, 23, and 27
- 6136 4. Sections 3.1.7 - Added this new section to clarify returning Unsupported Attributes for all
 6137 operations, including only returning attributes that were in the request. Issues 18, 23, and 27
- 6138 5. Sections 3.1.7 and 4.1 - clarified the encoding of the "out-of-band" 'unsupported' and 'unknown'
 6139 values. Issues 12 and 15
- 6140 6. Section 3.1.8 - clarified that only the version number parameter will be carried forward into
 6141 future major or minor versions of the protocol.
- 6142 7. Section 3.1.8 - indicated that IPP/1.1 Printers SHOULD support version '1.0' and that not all
 6143 previous minor versions need be supported. Issue 33
- 6144 8. Section 3.1.8 - relaxed the requirements to increment the major version number. Issue 33
- 6145 9. Section 3.1.9, and 3.2.5 - added the 'processing' state to the list of job states that a job can be in
 6146 after a Create-Job operation. Issue 13
- 6147 10. Section 3.1.9 - clarified that a non-spooling Printer MAY accept zero or more subsequent jobs
 6148 while processing a job and flow control them down. Subsequent create requests are rejected with
 6149 the 'server-error-busy' error status. Issue 20
- 6150 ~~12. Section 3.2.1.1 - clarified that the Printer object rejects a Print-Job request if it does not support~~
 6151 ~~the "compression" operation attribute and a client supplies it.~~
- 6152 11. Section 3.2.1.1 - clarified the validation of the "compression" operation attribute and its
 6153 relationship to the validation of the "document-format" attribute and returning Unsupported
 6154 Attributes. Issues 6, 11, and 28
- 6155 12. Sections 3.2.1.1, 4.3.8, 13.1.4.16, and 13.1.4.17 - added the 'client-error-compression-not-
 6156 supported', 'client-error-compression-error' status codes and the 'unsupported-compression' and
 6157 'compression-error' job-state-reasons. Issue 28

- 6158 13. Sections 3.2.1.1 and 4.3.8 - added 'unsupported-document-format' and 'document-format-error'
6159 job-state-reasons. Issue 3
- 6160 14. Sections 3.2.2, 4.3.8 and 13.1.4.19 - added 'client-error-document-access-error' status code and
6161 'document-access-error' job state reason. Issue 35
- 6162 15. Section 3.2.6 - clarified that "limit" takes precedence over "which-jobs" and "my-jobs". Issue 8
- 6163 16. Section 3.2.6.2 - clarified that Get-Jobs returns 'successful-ok' when no jobs to return. Issue 24
- 6164 17. Sections 3.2.7, 3.2.8, and 3.2.9 - added the OPTIONAL Pause-Printer, Resume-Printer, and
6165 Purge-Jobs operations
- 6166 18. Sections 3.3.5, 3.3.6, and 3.3.7 - added the OPTIONAL Hold-Job, Release-Job, and Restart-Job
6167 operations.
- 6168 ~~5. Section 4.1.9 - added 'image-tiff' and 'application/pdf' values.~~
- 6169 19. Section 4.1.9.1 - clarified that 'application/octet-stream' auto-sensing can happen at create request
6170 time and/or job/document processing time. Issue 9 and Issue 10
- 6171 20. Section 4.2 - clarified that xxx-supported have multiple keywords and/or names by adding
6172 parentheses to the table to give: (1setOf (type3 keyword | name))
- 6173 21. Section 4.2.2 - added the 'indefinite' keyword value to the "job-hold-until" attribute for use with
6174 the create operations and Hold-Job and Restart-Job operations.
- 6175 22. Section 4.2.4 - clarified that "multiple-document-handling" MUST be supported if the Printer
6176 supports multiple documents per job Issue 34
- 6177 23. Section 4.2.6 - added more enum values to the "finishings" Job Template attribute.
- 6178 24. Section 4.3.7 - added that a forwarding server that cannot get any job state MAY return the job's
6179 state as 'completed', provided that it also return the new 'queued-in-device' job state reason. Issue
6180 14
- 6181 25. Section 4.3.7.1 - added the Partitioning of Job States section.
- 6182 26. Section 4.3.8 - added 'job-data-insufficient' job state reason to indicate whether sufficient data
6183 has arrived for the document to start to be processed. Issue 13
- 6184 27. Section 4.3.8 - added 'document-access-error' job state reason to indicate an access error of any
6185 kind. Issue 35
- 6186 28. Section 4.3.8 - added 'job-queued-for-marker' job state reason to indicate whether the job has
6187 completed some processing and is waiting for the marker. Issue 31
- 6188 29. Section 4.3.8 - added 'unsupported-compression' and 'compression-error' job state reasons to
6189 indicate compression not supported or compression processing error after the create has been
6190 accepted. Issue 6
- 6191 30. Section 4.3.8 - added 'unsupported-document-format' and 'document-format-error' job state
6192 reasons to indicate document not supported or document format processing error after the create
6193 has been accepted. Issue 3
- 6194 31. Section 4.3.8 - added 'queued-in-device' job state reason to indicate that a job as been forwarded
6195 to a print system or device that does not provide any job status. Issue 14

- 6196 32. Section 4.4.3 - clarified "uri-security-supported" is orthogonal to Client Authentication so that
6197 'none' does not exclude Client Authentication. Issue 2
- 6198 33. Section 4.4.12 - added the 'moving-to-paused' keyword value to the "printer-state-reasons"
6199 attribute for use with the Pause-Job operation.
- 6200 34. Section 4.4.12 - replaced the duplicate 'marker-supply-low' keyword with the missing 'toner-
6201 empty' keyword for the "printer-state-reasons" attribute.
- 6202 35. Section 4.4.15 - added the enum values to the "operations-supported" attribute for the new
6203 operations. Clarified that the values of this attribute are encoded as any enum, namely 32-bit
6204 values.
- 6205 36. Sections 4.4.36 and 4.4.37 - added the OPTIONAL "pages-per-minute" and "pages-per-minute-
6206 color" Printer Description attributes.
- 6207 37. Section 4.4.30 - clarified that the dateTime value of "printer-current-time" is on a "best efforts
6208 basis". If a proper date-time cannot be obtained, the implementation returns the 'no-value' out-of-
6209 band value. Also clarified that the time zone NEED NOT be the time zone that the people near
6210 the device use and that the client SHOULD display the dateTime attributes in the user's local
6211 time. Issue 17
- 6212 38. Section 5.1 - clarified that any response MAY contain additional attribute groups, attributes, or
6213 attribute values. Issues 25 and 26
- 6214 39. Section 5.1 - clarified that a client SHOULD do its best to prevent a channel from being closed
6215 by a lower layer when the channel is flow controlled off by the IPP Printer. Issues 4 and 5
- 6216 40. Section 8.3 - clarified the use of URIs for each Client Authentication mechanism.
- 6217 41. Section 8.5 - added the security discussion around the new operator operations.
- 6218 42. Section 13.1.4.16 - added client-error-compression-not-supported (0x040F) Issue 6
- 6219 43. Section 13.1.4.17 - added client-error-compression-error (0x0410) Issue 6
- 6220 44. Section 13.1.4.18 - added client-error-document-format-error (0x0411) Issue 28
- 6221 45. Section 13.1.4.19 - added client-error-document-access-error (0x0412) Issue 35
- 6222 46. Section 13.1.5.10 - added server-error-multiple-document-jobs-not-supported (0x0509) Issue 34
- 6223 47. Section 16 - added the OPTIONAL "pages-per-minute" and "pages-per-minute-color" Printer
6224 attributes to the Directory schema.
- 6225 48. Section 16 - added OPTIONAL "multiple-document-jobs-supported" to the Directory schema.
6226 Issue 34
- 6227 49. Section 16 - added RECOMMENDED "uri-security-supported", "compression-supported", and
6228 "ipp-versions-supported" to the Directory schema. Issues 2,

6229 The following changes in semantics and/or conformance have been incorporated into ~~were made to~~
6230 IPP/1.0 [IPP-MOD1.0] to create this IPP/1.1 document:

- 6231 1. Section 3.1.8, 5.2.4, and 13.1.5.4 - Clients and IPP objects MUST support ~~both~~ version 1.0 and
6232 1.1 and SHOULD support version 1.0. Clients MUST support version 1.1 and MAY support
6233 version 1.0. Issue 33 and Issue 36
- 6234 2. Section 3.2.1.1 and section 4.4.32 - changed the "compression" and "compression-supported"
6235 attributes from OPTIONAL to REQUIRED. Issue 28
- 6236 3. Sections 3.2.1.2 and 4.3.8 - changed "job-state-reasons" from RECOMMENDED to
6237 REQUIRED, so that "job-state-reasons" MUST be returned in create operation responses. Issue
6238 30

- 6239 4. Sections 3.2.4, 3.3.1, 4.4.16, and 16 - changed Create-Job/Send-Document so that they MAY be
6240 implemented while only supporting one document jobs. Added the "multiple-document-jobs-
6241 supported" boolean Printer Description attribute to indicate whether Create-Job/Send-Document
6242 support multiple document jobs or not. Added to the Directory schema. Issue 34
- 6243 5. Section 4.1.9 - deleted 'text/plain; charset=iso-10646-ucs-2', since binary is not legal with the
6244 'text' type.
- 6245 6. Section 4.3.8 - changed "job-state-reasons" from RECOMMENDED to REQUIRED. Issue 30
- 6246 7. Section 4.3.12 - added OPTIONAL 'dateTime' attribute syntax to "time-at-creation", "time-at-
6247 processing", and "time-at-completed" Job Description attributes for use in version '1.1' responses.
6248 Issue 17
- 6249 8. Section 4.3.12 - changed the "time-at-creation", "time-at-processing", and "time-at-completed"
6250 Event Time Job Description attributes from OPTIONAL to REQUIRED. Issue 17
- 6251 9. Section 4.3.12.4 - added the REQUIRED "job-printer-up-time (integer(1:MAX))" Job
6252 Description attribute as an alias for "printer-up-time" to reduce number of operations to get job
6253 times. Issue 17
- 6254 10. Section 4.4.2 - added the REQUIRED "uri-authentication-supported (1setOf type2 keyword)"
6255 Printer Description attribute to describe the Client Authentication used by each Printer URI.
6256 Issue 2
- 6257 11. Section 4.4.11 - clarified the "printer-state" to allow a Printer that can interpret one or more jobs
6258 (rip) while marking one job to have those jobs all in the 'processing' state. Issue 31
- 6259 12. Section 4.4.12 - changed "printer-state-reasons" Printer Description attribute from OPTIONAL to
6260 REQUIRED. Issue 30
- 6261 13. Section 4.4.14 - added the REQUIRED "ipp-versions-supported (1setOf keyword)" Printer
6262 Description attribute, since IPP/1.1 Printers do not have to support version '1.0'.
- 6263 14. Section 4.4.16 - added the REQUIRED "multiple-document-jobs-supported (boolean)" Printer
6264 Description attribute so that a client can tell whether a Printer that supports Create-Job/Send-
6265 Document supports multiple document jobs or not. Issue 34
- 6266 15. Section 4.4.24 - changed the "queued-job-count" Printer Description attribute from
6267 RECOMMENDED to REQUIRED. Issue 29
- 6268 16. Section 4.4.32 - changed "compression-supported (1setOf type3 keyword)" Printer Description
6269 attribute from OPTIONAL to REQUIRED. Issue 28
- 6270 17. Section 5.1, ~~5.2, and 8.2 - changed the IPP object security requirements from OPTIONAL non-~~
6271 ~~standards track SSL3 to RECOMMENDED standards track TLS.~~ changed the client security
6272 requirements from RECOMMENDED non-standards track SSL3 to MUST/SHOULD [which is
6273 to be determined in consultation with the Area Director] support Client Authentication as defined
6274 in the IPP/1.1 Encoding and Transport document [IPP-PRO]. A client SHOULD support
6275 Operation Privacy and Server Authentication as defined in the IPP/1.1 Encoding and Transport
6276 document [IPP-PRO]. ~~RECOMMENDED standards track TLS~~
- 6277 18. Section 5.2.7 - changed the IPP object security requirements from OPTIONAL non-standards
6278 track SSL3 to MUST/SHOULD [which is to be determined in consultation with the Area
6279 Director] contain support for Client Authentication as defined in the IPP/1.1 Encoding and
6280 Transport document [IPP-PRO]. A Printer implementation MAY allow an administrator to
6281 configure the Printer so that all, some, or none of the users are authenticated. An IPP Printer
6282 implementation SHOULD contain support for Operation Privacy and Server Authentication as
6283 defined in the IPP/1.1 Encoding and Transport document [IPP-PRO]. A Printer implementation

6284 MAY allow an administrator to configure the degree of support for Operation Privacy and Server
6285 Authentication.

6286 See also the "IPP/1.1 Encoding and Transport" [IPP-PRO] document for differences between IPP/1.0
6287 [[RFC2565IPP-PRO1.0](#)] and IPP/1.1 [IPP-PRO].

6288 18. Full Copyright Statement

6289 Copyright (C) The Internet Society (1999). All Rights Reserved.

6290 This document and translations of it may be copied and furnished to others, and derivative works that
6291 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published
6292 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright
6293 notice and this paragraph are included on all such copies and derivative works. However, this document
6294 itself may not be modified in any way, such as by removing the copyright notice or references to the
6295 Internet Society or other Internet organizations, except as needed for the purpose of developing Internet
6296 standards in which case the procedures for copyrights defined in the Internet Standards process must be
6297 followed, or as required to translate it into languages other than English.

6298 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or
6299 its successors or assigns.

6300 This document and the information contained herein is provided on an "AS IS" basis and THE
6301 INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL
6302 WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
6303 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY
6304 RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
6305 PARTICULAR PURPOSE.

6306