

1 INTERNET-DRAFT
2 draft-ietf-ipp-model-v11-~~0204~~.txt

R. deBry
IBM Corporation
T. Hastings (editor)
Xerox Corporation
R. Herriot
Xerox Corporation
S. Isaacson
Novell, Inc.
P. Powell
Astart Technologies
~~February 17~~April 6, 1998

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 [IPP-REQ]

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

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

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

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

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

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

67

Table of Contents

68	1. Introduction	10
69	1.1 Simplified Printing Model	11
70	2. IPP Objects	13
71	2.1 Printer Object	13
72	2.2 Job Object	15
73	2.3 Object Relationships	16
74	2.4 Object Identity	17
75	3. IPP Operations	19
76	3.1 Common Semantics	20
77	3.1.1 Required Parameters	20
78	3.1.2 Operation IDs and Request IDs	20
79	3.1.3 Attributes	21
80	3.1.4 Character Set and Natural Language Operation Attributes	22
81	3.1.4.1 Request Operation Attributes	23
82	3.1.4.2 Response Operation Attributes	25
83	3.1.5 Operation Targets	26
84	3.1.6 Operation Status Codes and Messages	28
85	3.1.7 Versions	28
86	3.1.8 Job Creation Operations	30
87	3.2 Printer Operations	31
88	3.2.1 Print-Job Operation	32
89	3.2.1.1 Print-Job Request	32
90	3.2.1.2 Print-Job Response	35
91	3.2.2 Print-URI Operation	38
92	3.2.3 Validate-Job Operation	38
93	3.2.4 Create-Job Operation	38
94	3.2.5 Get-Printer-Attributes Operation	39
95	3.2.5.1 Get-Printer-Attributes Request	39
96	3.2.5.2 Get-Printer-Attributes Response	41
97	3.2.6 Get-Jobs Operation	42
98	3.2.6.1 Get-Jobs Request	42
99	3.2.6.2 Get-Jobs Response	44
100	3.2.7 Pause-Printer Operation	45
101	3.2.7.1 Pause-Printer Request	46
102	3.2.7.2 Pause-Printer Response	46
103	3.2.8 Resume-Printer Operation	47
104	3.2.9 Purge-Jobs Operation	48
105	3.3 Job Operations	48
106	3.3.1 Send-Document Operation	48
107	3.3.1.1 Send-Document Request	49
108	3.3.1.2 Send-Document Response	51

109	3.3.2	Send-URI Operation	51
110	3.3.3	Cancel-Job Operation	52
111	3.3.3.1	Cancel-Job Request	52
112	3.3.3.2	Cancel-Job Response	52
113	3.3.4	Get-Job-Attributes Operation	53
114	3.3.4.1	Get-Job-Attributes Request	54
115	3.3.4.2	Get-Job-Attributes Response	54
116	3.3.5	Hold-Job Operation	55
117	3.3.5.1	Hold-Job Request	56
118	3.3.5.2	Hold-Job Response	57
119	3.3.6	Release-Job Operation	57
120	3.3.7	Restart-Job Operation	58
121	3.3.7.1	Restart-Job Request	59
122	3.3.7.2	Restart-Job Response	60
123	4.	Object Attributes	60
124	4.1	Attribute Syntaxes	60
125	4.1.1	'text'	61
126	4.1.1.1	'textWithoutLanguage'	62
127	4.1.1.2	'textWithLanguage'	62
128	4.1.2	'name'	62
129	4.1.2.1	'nameWithoutLanguage'	63
130	4.1.2.2	'nameWithLanguage'	63
131	4.1.2.3	Matching 'name' attribute values	63
132	4.1.3	'keyword'	64
133	4.1.4	'enum'	65
134	4.1.5	'uri'	65
135	4.1.6	'uriScheme'	65
136	4.1.7	'charset'	66
137	4.1.8	'naturalLanguage'	66
138	4.1.9	'mimeType'	67
139	4.1.10	'octetString'	68
140	4.1.11	'boolean'	68
141	4.1.12	'integer'	68
142	4.1.13	'rangeOfInteger'	68
143	4.1.14	'dateTime'	69
144	4.1.15	'resolution'	69
145	4.1.16	'1setOf X'	69
146	4.2	Job Template Attributes	69
147	4.2.1	job-priority (integer(1:100))	73
148	4.2.2	job-hold-until (type3 keyword name (MAX))	74
149	4.2.3	job-sheets (type3 keyword name(MAX))	74
150	4.2.4	multiple-document-handling (type2 keyword)	75
151	4.2.5	copies (integer(1:MAX))	76
152	4.2.6	finishings (1setOf type2 enum)	76

153	4.2.7	page-ranges (1setOf rangeOfInteger (1:MAX))	78
154	4.2.8	sides (type2 keyword).....	79
155	4.2.9	number-up (integer(1:MAX)).....	79
156	4.2.10	orientation-requested (type2 enum).....	80
157	4.2.11	media (type3 keyword name(MAX)).....	81
158	4.2.12	printer-resolution (resolution)	81
159	4.2.13	print-quality (type2 enum).....	81
160	4.3	Job Description Attributes	82
161	4.3.1	job-uri (uri)	84
162	4.3.2	job-id (integer(1:MAX)).....	84
163	4.3.3	job-printer-uri (uri)	84
164	4.3.4	job-more-info (uri).....	84
165	4.3.5	job-name (name(MAX)).....	85
166	4.3.6	job-originating-user-name (name(MAX)).....	85
167	4.3.7	job-state (type1 enum).....	85
168	4.3.7.1	Partitioning of Job States	88
169	4.3.8	job-state-reasons (1setOf type2 keyword)	88
170	4.3.9	job-state-message (text(MAX)).....	91
171	4.3.10	number-of-documents (integer(0:MAX)).....	91
172	4.3.11	output-device-assigned (name(127))	91
173	4.3.12	time-at-creation (integer(0:MAX)).....	91
174	4.3.13	time-at-processing (integer(0:MAX)).....	92
175	4.3.14	time-at-completed (integer(0:MAX))	92
176	4.3.15	number-of-intervening-jobs (integer(0:MAX)).....	92
177	4.3.16	job-message-from-operator (text(127))	92
178	4.3.17	job-k-octets (integer(0:MAX))	92
179	4.3.18	job-impressions (integer(0:MAX)).....	93
180	4.3.19	job-media-sheets (integer(0:MAX))	93
181	4.3.20	job-k-octets-processed (integer(0:MAX))	93
182	4.3.21	job-impressions-completed (integer(0:MAX)).....	94
183	4.3.22	job-media-sheets-completed (integer(0:MAX))	94
184	4.3.23	attributes-charset (charset).....	94
185	4.3.24	attributes-natural-language (naturalLanguage).....	94
186	4.4	Printer Description Attributes.....	94
187	4.4.1	printer-uri-supported (1setOf uri).....	97
188	4.4.2	uri-security-supported (1setOf type2 keyword).....	97
189	4.4.3	printer-name (name(127)).....	99
190	4.4.4	printer-location (text(127))	99
191	4.4.5	printer-info (text(127))	99
192	4.4.6	printer-more-info (uri)	99
193	4.4.7	printer-driver-installer (uri)	99
194	4.4.8	printer-make-and-model (text(127)).....	99
195	4.4.9	printer-more-info-manufacturer (uri)	99
196	4.4.10	printer-state (type1 enum)	100
197	4.4.11	printer-state-reasons (1setOf type2 keyword).....	101

198	4.4.12	printer-state-message (text(MAX))	103
199	4.4.13	operations-supported (1setOf type2 enum)	103
200	4.4.14	charset-configured (charset)	104
201	4.4.15	charset-supported (1setOf charset)	104
202	4.4.16	natural-language-configured (naturalLanguage)	104
203	4.4.17	generated-natural-language-supported (1setOf naturalLanguage).....	104
204	4.4.18	document-format-default (mimeMediaType).....	105
205	4.4.19	document-format-supported (1setOf mimeMediaType).....	105
206	4.4.20	printer-is-accepting-jobs (boolean).....	105
207	4.4.21	queued-job-count (integer(0:MAX))	105
208	4.4.22	printer-message-from-operator (text(127)).....	106
209	4.4.23	color-supported (boolean)	106
210	4.4.24	reference-uri-schemes-supported (1setOf uriScheme)	106
211	4.4.25	pdl-override-supported (type2 keyword).....	106
212	4.4.26	printer-up-time (integer(1:MAX)).....	106
213	4.4.27	printer-current-time (dateTime).....	107
214	4.4.28	multiple-operation-time-out (integer(1:MAX)).....	107
215	4.4.29	compression-supported (1setOf type3 keyword).....	107
216	4.4.30	job-k-octets-supported (rangeOfInteger(0:MAX))	108
217	4.4.31	job-impressions-supported (rangeOfInteger(0:MAX)).....	108
218	4.4.32	job-media-sheets-supported (rangeOfInteger(0:MAX)).....	108
219	4.4.33	pages-per-minute (integer(0:MAX))	108
220	4.4.34	pages-per-minute-color (integer(0:MAX)).....	108
221	5.	Conformance	109
222	5.1	Client Conformance Requirements.....	109
223	5.2	IPP Object Conformance Requirements	110
224	5.2.1	Objects.....	110
225	5.2.2	Operations.....	110
226	5.2.3	IPP Object Attributes.....	111
227	5.2.4	Versions.....	111
228	5.2.5	Extensions	111
229	5.2.6	Attribute Syntaxes	111
230	5.3	Charset and Natural Language Requirements.....	112
231	5.4	Security Conformance Requirements	112
232	6.	IANA Considerations (registered and private extensions).....	112
233	6.1	Typed 'keyword' and 'enum' Extensions	113
234	6.2	Attribute Extensibility	115
235	6.3	Attribute Syntax Extensibility	115
236	6.4	Operation Extensibility.....	115
237	6.5	Attribute Groups	116
238	6.6	Status Code Extensibility.....	116
239	6.7	Registration of MIME types/sub-types for document-formats	117
240	6.8	Registration of charsets for use in 'charset' attribute values	117

241	7.	Internationalization Considerations.....	117
242	8.	Security Considerations.....	119
243	8.1	Security Scenarios.....	120
244	8.1.1	Client and Server in the Same Security Domain.....	121
245	8.1.2	Client and Server in Different Security Domains.....	121
246	8.1.3	Print by Reference.....	121
247	8.2	URIs for TLS and non-TLS Access.....	121
248	8.3	The "requesting-user-name" (name(MAX)) Operation Attribute.....	121
249	8.4	Restricted Queries.....	123
250	8.5	Operations performed by operators and system administrators.....	123
251	8.6	Queries on jobs submitted using non-IPP protocols.....	123
252	8.7	IPP Security Application Profile for TLS.....	124
253	9.	See the TLS specification [RFC2246] for a discussion of any government export restrictions on implementations conforming to the "Mandatory Cipher Suites". References.....	124
254			
255	10.	Notices	127
256	11.	Author's Address.....	128
257	12.	Formats for IPP Registration Proposals.....	132
258	12.1	Type2 keyword attribute values registration.....	132
259	12.2	Type3 keyword attribute values registration.....	132
260	12.3	Type2 enum attribute values registration.....	132
261	12.4	Type3 enum attribute values registration.....	133
262	12.5	Attribute registration.....	133
263	12.6	Attribute Syntax registration.....	134
264	12.7	Operation registration.....	134
265	12.8	Attribute Group registration.....	134
266	12.9	Status code registration.....	135
267	13.	APPENDIX A: Terminology.....	135
268	13.1	Conformance Terminology.....	135
269	13.1.1	NEED NOT.....	135
270	13.2	Model Terminology.....	135
271	13.2.1	Keyword.....	135
272	13.2.2	Attributes.....	135
273	13.2.2.1	Attribute Name.....	136
274	13.2.2.2	Attribute Group Name.....	136
275	13.2.2.3	Attribute Value.....	136
276	13.2.2.4	Attribute Syntax.....	136
277	13.2.3	Supports.....	136
278	13.2.4	print-stream page.....	138
279	13.2.5	impression.....	138

280	14. APPENDIX B: Status Codes and Suggested Status Code Messages.....	138
281	14.1 Status Codes	139
282	14.1.1 Informational	139
283	14.1.2 Successful Status Codes	139
284	14.1.2.1 successful-ok (0x0000).....	139
285	14.1.2.2 successful-ok-ignored-or-substituted-attributes (0x0001).....	140
286	14.1.2.3 successful-ok-conflicting-attributes (0x0002).....	140
287	14.1.3 Redirection Status Codes.....	140
288	14.1.4 Client Error Status Codes	140
289	14.1.4.1 client-error-bad-request (0x0400).....	140
290	14.1.4.2 client-error-forbidden (0x0401).....	141
291	14.1.4.3 client-error-not-authenticated (0x0402).....	141
292	14.1.4.4 client-error-not-authorized (0x0403).....	141
293	14.1.4.5 client-error-not-possible (0x0404).....	141
294	14.1.4.6 client-error-timeout (0x0405).....	141
295	14.1.4.7 client-error-not-found (0x0406).....	141
296	14.1.4.8 client-error-gone (0x0407).....	142
297	14.1.4.9 client-error-request-entity-too-large (0x0408).....	142
298	14.1.4.10 client-error-request-value-too-long (0x0409).....	142
299	14.1.4.11 client-error-document-format-not-supported (0x040A).....	143
300	14.1.4.12 client-error-attributes-or-values-not-supported (0x040B).....	143
301	14.1.4.13 client-error-uri-scheme-not-supported (0x040C).....	143
302	14.1.4.14 client-error-charset-not-supported (0x040D).....	143
303	14.1.4.15 client-error-conflicting-attributes (0x040E).....	143
304	14.1.4.16 <u>client-error-compression-not-supported (0x040F).....</u>	143
305	14.1.5 Server Error Status Codes	144
306	14.1.5.1 server-error-internal-error (0x0500).....	144
307	14.1.5.2 server-error-operation-not-supported (0x0501).....	144
308	14.1.5.3 server-error-service-unavailable (0x0502).....	144
309	14.1.5.4 server-error-version-not-supported (0x0503).....	144
310	14.1.5.5 server-error-device-error (0x0504).....	145
311	14.1.5.6 server-error-temporary-error (0x0505).....	145
312	14.1.5.7 server-error-not-accepting-jobs (0x0506).....	145
313	14.1.5.8 server-error-busy (0x0507).....	145
314	14.1.5.9 server-error-job-canceled (0x0508).....	145
315	14.2 Status Codes for IPP Operations.....	146
316	15. APPENDIX C: "media" keyword values	148
317	16. APPENDIX D: Processing IPP Attributes	151
318	16.1 Fidelity 152	
319	16.2 Page Description Language (PDL) Override.....	153
320	16.3 Using Job Template Attributes During Document Processing.....	154
321	17. APPENDIX E: Generic Directory Schema	155

322 18. APPENDIX F: Differences between the IPP/1.0 and IPP/1.1 "Model and Semantics" Specifications
323 158
324

325 1. Introduction

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

- 330 Design Goals for an Internet Printing Protocol [IPP-REQ]
- 331 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [IPP-RAT]
- 332 Internet Printing Protocol/1.1: Model and Semantics (this document)
- 333 Internet Printing Protocol/1.1: Encoding and Transport [IPP-PRO]
- 334 Internet Printing Protocol/1.1: Implementer's Guide [IPP-IIG]
- 335 Mapping between LPD and IPP Protocols [IPP-LPD]

336

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

339 This document is laid out as follows:

- 340 - The rest of Section 1 is an introduction to the IPP simplified model for distributed printing.
- 341 - Section 2 introduces the object types covered in the model with their basic behaviors, attributes,
342 and interactions.
- 343 - Section 3 defines the operations included in IPP/1.1. IPP operations are synchronous, therefore, for
344 each operation, there is a both request and a response.
- 345 - Section 4 defines the attributes (and their syntaxes) that are used in the model.
- 346 - Sections 5 - 6 summarizes the implementation conformance requirements for objects that support
347 the protocol and IANA considerations, respectively.
- 348 - Sections 7 - 12 cover the Internationalization and Security considerations as well as References,
349 Intellectual Property Notice, Copyright Notice, Author contact information, and Formats for
350 Registration Proposals.
- 351 - Sections 13 - 15 are appendices that cover Terminology, Status Codes and Messages, and "media"
352 keyword values.

353 Note: This document uses terms such as "attributes", "keywords", and "support". These
354 terms have special meaning and are defined in the model terminology section 13.2.
355 Capitalized terms, such as MUST, MUST NOT, REQUIRED, SHOULD, SHOULD
356 NOT, MAY, NEED NOT, and OPTIONAL, have special meaning relating to
357 conformance. These terms are defined in section 13.1 on conformance terminology, most
358 of which is taken from RFC 2119 [RFC2119].

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

364 - Section 18 is an appendix summarizing the additions and changes from the IPP/1.0 "Model and
365 Semantics" specification [IPP-MOD1.0] to make this IPP/1.1 document.

366 1.1 Simplified Printing Model

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

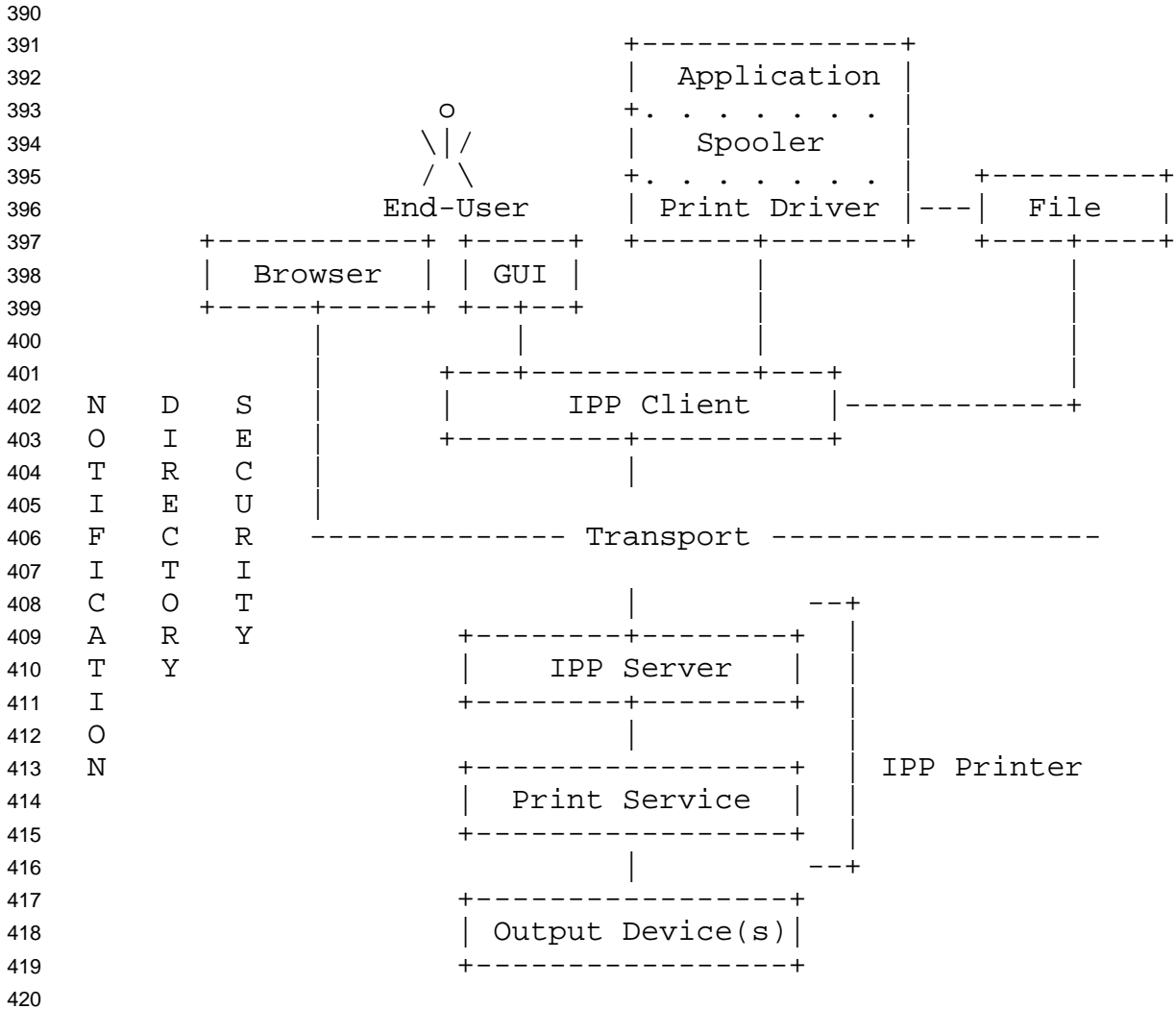
381 The IPP/1.1 model encapsulates the important components of distributed printing into two object types:

- 382 - Printer (Section 2.1)
- 383 - Job (Section 2.2)

384

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

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



421 An IPP Printer object encapsulates the functions normally associated with physical output devices along
422 with the spooling, scheduling and multiple device management functions often associated with a print
423 server. Printer objects are optionally registered as entries in a directory where end users find and select
424 them based on some sort of filtered and context based searching mechanism (see section 17). The
425 directory is used to store relatively static information about the Printer, allowing end users to search for
426 and find Printers that match their search criteria, for example: name, context, printer capabilities, etc.
427 The more dynamic information, such as state, currently loaded and ready media, number of jobs at the
428 Printer, errors, warnings, and so forth, is directly associated with the Printer object itself rather than with
429 the entry in the directory which only represents the Printer object.

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

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

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

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

451 2. IPP Objects

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

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

- 461 - "REQUIRED": each object MUST support the attribute.
- 462 - "OPTIONAL": each object MAY support the attribute.

463

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

466 2.1 Printer Object

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

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

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

- 475 - "job-template" attributes: These attributes describe supported job processing capabilities and
476 defaults for the Printer object. (See section 4.2)
 - 477 - "printer-description" attributes: These attributes describe the Printer object's identification, state,
478 location, references to other sources of information about the Printer object, etc. (see section 4.4)
- 479

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

483 Some examples of configurations supporting a Printer object include:

- 484 1) An output device with no spooling capabilities
 - 485 2) An output device with a built-in spooler
 - 486 3) A print server supporting IPP with one or more associated output devices
 - 487 3a) The associated output devices may or may not be capable of spooling jobs
 - 488 3b) The associated output devices may or may not support IPP
- 489

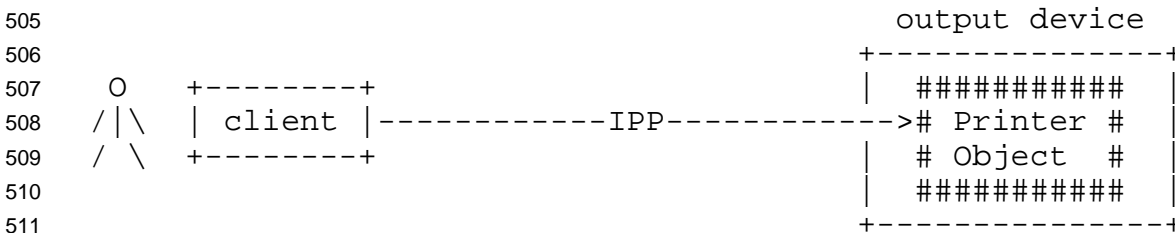
490 The following figures show some examples of how Printer objects can be realized on top of various
491 distributed printing configurations. The embedded case below represents configurations 1 and 2. The
492 hosted and fan-out figures below represent configurations 3a and 3b.

493 Legend:

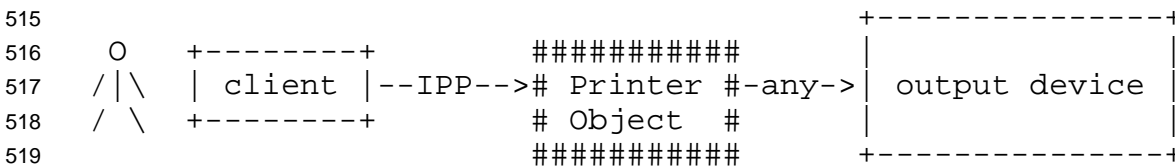
494
495 ##### indicates a Printer object which is
496 either embedded in an output device or is
497 hosted in a server. The Printer object
498 might or might not be capable of queuing/spooling.

499
500 any indicates any network protocol or direct
501 connect, including IPP

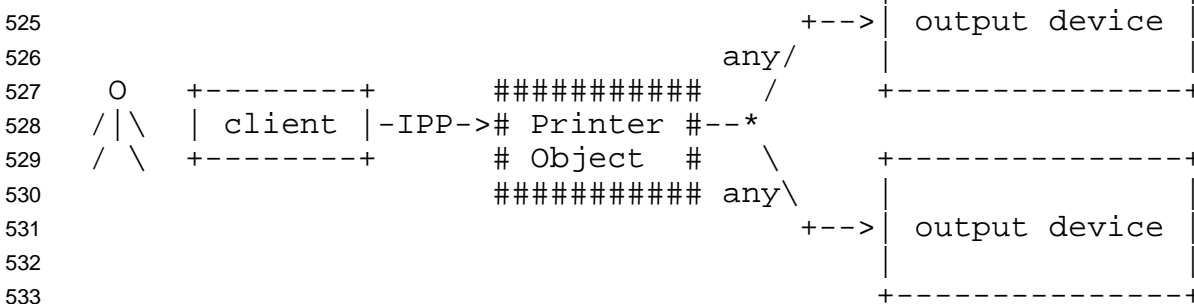
502
503
504 embedded printer:



512
513
514 hosted printer:



523
524 fan out:



536 2.2 Job Object

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

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

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

- 543 - "job-template" attributes: These attributes can be supplied by the client or end user and include job
544 processing instructions which are intended to override any Printer object defaults and/or
545 instructions embedded within the document data. (See section 4.2)
- 546 - "job-description" attributes: These attributes describe the Job object's identification, state, size, etc.
547 The client supplies some of these attributes, and the Printer object generates others. (See section
548 4.3)

549
550 An implementation **MUST** support at least one document per Job object. An implementation **MAY**
551 support multiple documents per Job object. A document is either:

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

555

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

559 2.3 Object Relationships

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

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

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

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

575 2.4 Object Identity

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

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

589 If a Printer object supports more than one communication channel, some or all of those channels might
590 support and/or require different security mechanisms. In such cases, an administrator could expose the
591 simultaneous support for these multiple communication channels as multiple URIs for a single Printer
592 object where each URI represents one of the communication channels to the Printer object. To support
593 this flexibility, the IPP Printer object type defines a multi-valued identification attribute called the
594 "printer-uri-supported" attribute. It MUST contain at least one URI. It MAY contain more than one
595 URI. That is, every Printer object will have at least one URI that identifies at least one communication
596 channel to the Printer object, but it may have more than one URI where each URI identifies a different
597 communication channel to the Printer object. The "printer-uri-supported" attribute has a companion
598 attribute, the "uri-security-supported" attribute, that has the same cardinality as "printer-uri-supported".
599 The purpose of the "uri-security-supported" attribute is to indicate the security mechanisms (if any) used
600 for each URI listed in "printer-uri-supported". These two attributes are fully described in sections 4.4.1
601 and 4.4.2.

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

605 Note: IPP/1.1 does not specify how the client obtains the client supplied URI, but it is
606 RECOMMENDED that a Printer object be registered as an entry in a directory service. End-users and
607 programs can then interrogate the directory searching for Printers. Section 17 defines a generic schema
608 for Printer object entries in the directory service and describes how the entry acts as a bridge to the actual
609 IPP Printer object. The entry in the directory that represents the IPP Printer object includes the possibly
610 many URIs for that Printer object as values in one its attributes.

611 When a client submits a create request to the Printer object, the Printer object validates the request and
612 creates a new Job object. The Printer object assigns the new Job object a URI which is stored in the
613 "job-uri" Job attribute. This URI is then used by clients as the target for subsequent Job operations. The

614 Printer object generates a Job URI based on its configured security policy and the URI used by the client
615 in the create request.

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

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

628 Allowing Job objects to have URIs allows for flexibility and scalability. For example, in some
629 implementations, the Printer object might create Jobs that are processed in the same local environment
630 as the Printer object itself. In this case, the Job URI might just be a composition of the Printer's URI and
631 some unique component for the Job object, such as the unique 32-bit positive integer mentioned later in
632 this paragraph. In other implementations, the Printer object might be a central clearing-house for
633 validating all Job object creation requests, but the Job object itself might be created in some environment
634 that is remote from the Printer object. In this case, the Job object's URI may have no physical-location
635 relationship at all to the Printer object's URI. Again, the fact that Job objects have URIs allows for
636 flexibility and scalability, however, many existing printing systems have local models or interface
637 constraints that force print jobs to be identified using only a 32-bit positive integer rather than an
638 independent URI. This numeric Job ID is only unique within the context of the Printer object to which
639 the create request was originally submitted. Therefore, in order to allow both types of client access to
640 IPP Job objects (either by Job URI or by numeric Job ID), when the Printer object successfully processes
641 a create request and creates a new Job object, the Printer object MUST generate both a Job URI and a
642 Job ID. The Job ID (stored in the "job-id" attribute) only has meaning in the context of the Printer object
643 to which the create request was originally submitted. This requirement to support both Job URIs and Job
644 IDs allows all types of clients to access Printer objects and Job objects no matter the local constraints
645 imposed on the client implementation.

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

652 To summarize:

- 653 - Each Printer object is identified with one or more URIs. The Printer's "printer-uri-supported"
654 attribute contains the URI(s).

- 655 - The Printer object's "uri-security-supported" attribute identifies the communication channel security
656 protocols that may or may not have been configured for the various Printer object URIs (e.g., 'tls'
657 or 'none').
- 658 - Each Job object is identified with a Job URI. The Job's "job-uri" attribute contains the URI.
- 659 - Each Job object is also identified with Job ID which is a 32-bit, positive integer. The Job's "job-id"
660 attribute contains the Job ID. The Job ID is only unique within the context of the Printer object
661 which created the Job object.
- 662 - Each Job object has a "job-printer-uri" attribute which contains the URI of the Printer object that
663 was used to create the Job object. This attribute is used to determine the Printer object that
664 created a Job object when given only the URI for the Job object. This linkage is necessary to
665 determine the languages, charsets, and operations which are supported on that Job (the basis for
666 such support comes from the creating Printer object).
- 667 - Each Printer object has a name (which is not necessarily unique). The administrator chooses and
668 sets this name through some mechanism outside the scope of this IPP/1.1 document. The Printer
669 object's "printer-name" attribute contains the name.
- 670 - Each Job object has a name (which is not necessarily unique). The client optionally supplies this
671 name in the create request. If the client does not supply this name, the Printer object generates a
672 name for the Job object. The Job object's "job-name" attribute contains the name.

673 3. IPP Operations

674 IPP objects support operations. An operation consists of a request and a response. When a client
675 communicates with an IPP object, the client issues an operation request to the URI for that object.
676 Operation requests and responses have parameters that identify the operation. Operations also have
677 attributes that affect the run-time characteristics of the operation (the intended target, localization
678 information, etc.). These operation-specific attributes are called operation attributes (as compared to
679 object attributes such as Printer object attributes or Job object attributes). Each request carries along
680 with it any operation attributes, object attributes, and/or document data required to perform the
681 operation. Each request requires a response from the object. Each response indicates success or failure
682 of the operation with a status code as a response parameter. The response contains any operation
683 attributes, object attributes, and/or status messages generated during the execution of the operation
684 request.

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

687 The IPP/1.1 Printer operations are:

- 688 Print-Job (section 3.2.1)
- 689 Print-URI (section 3.2.2)
- 690 Validate-Job (section 3.2.3)
- 691 Create-Job (section 3.2.4)
- 692 Get-Printer-Attributes (section 3.2.5)
- 693 Get-Jobs (section 3.2.6)
- 694 Pause-Printer (section 3.3.5)

695 Resume-Printer (section 3.3.6)

696 Purge-Jobs (section 3.3.7)

697

698 The Job operations are:

699 Send-Document (section 3.3.1)

700 Send-URI (section 3.3.2)

701 Cancel-Job (section 3.3.3)

702 Get-Job-Attributes (section 3.3.4)

703 Hold-Job (section 3.3.5)

704 Release-Job (section 3.3.6)

705 Restart-Job (section 3.3.7)

706

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

709 3.1 Common Semantics

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

712 3.1.1 Required Parameters

713 Every operation request contains the following REQUIRED parameters:

714 - a "version-number",

715 - an "operation-id",

716 - a "request-id", and

717 - the attributes that are REQUIRED for that type of request.

718

719 Every operation response contains the following REQUIRED parameters:

720 - a "version-number",

721 - a "status-code",

722 - the "request-id" that was supplied in the corresponding request, and

723 - the attributes that are REQUIRED for that type of response.

724

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

728 3.1.2 Operation IDs and Request IDs

729 Each IPP operation request includes an identifying "operation-id" value. Valid values are defined in the
730 "operations-supported" Printer attribute section (see section 4.4.13). The client specifies which
731 operation is being requested by supplying the correct "operation-id" value.

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

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

744 3.1.3 Attributes

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

- 747 - Operation Attributes: These attributes are passed in the operation and affect the IPP object's
748 behavior while processing the operation request and may affect other attributes or groups of
749 attributes. Some operation attributes describe the document data associated with the print job
750 and are associated with new Job objects, however most operation attributes do not persist beyond
751 the life of the operation. The description of each operation attribute includes conformance
752 statements indicating which operation attributes are REQUIRED and which are OPTIONAL for
753 an IPP object to support and which attributes a client MUST supply in a request and an IPP
754 object MUST supply in a response.
- 755 - Job Template Attributes: These attributes affect the processing of a job. A client OPTIONALLY
756 supplies Job Template Attributes in a create request, and the receiving object MUST be prepared
757 to receive all supported attributes. The Job object can later be queried to find out what Job
758 Template attributes were originally requested in the create request, and such attributes are
759 returned in the response as Job Object Attributes. The Printer object can be queried about its Job
760 Template attributes to find out what type of job processing capabilities are supported and/or what
761 the default job processing behaviors are, though such attributes are returned in the response as
762 Printer Object Attributes. The "ipp-attribute-fidelity" operation attribute affects processing of all
763 client-supplied Job Template attributes (see sections 3.2.1.2 and 16 for a full description of "ipp-
764 attribute-fidelity" and its relationship to other attributes).
- 765 - Job Object Attributes: These attributes are returned in response to a query operation directed at a
766 Job object.
- 767 - Printer Object Attributes: These attributes are returned in response to a query operation directed at a
768 Printer object.
- 769 - Unsupported Attributes: In a create request, the client supplies a set of Operation and Job Template
770 attributes. If any of these attributes or their values is unsupported by the Printer object, the
771 Printer object returns the set of unsupported attributes in the response. Sections 3.2.1.2 and 16
772 give a full description of how Job Template attributes supplied by the client in a create request
773 are processed by the Printer object and how unsupported attributes are returned to the client.

774 Because of extensibility, any IPP object might receive a request that contains new or unknown
775 attributes or values for which it has no support. In such cases, the IPP object processes what it
776 can and returns the unsupported attributes in the response.
777

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

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

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

790 Note: Some operations are REQUIRED for IPP objects to support; the others are OPTIONAL (see
791 section 5.2.2). Therefore, before using an OPTIONAL operation, a client SHOULD first use the
792 REQUIRED Get-Printer-Attributes operation to query the Printer's "operations-supported" attribute in
793 order to determine which OPTIONAL Printer and Job operations are actually supported. The client
794 SHOULD NOT use an OPTIONAL operation that is not supported. When an IPP object receives a
795 request to perform an operation it does not support, it returns the 'server-error-operation-not-supported'
796 status code (see section 14.1.5.2). An IPP object is non-conformant if it does not support a REQUIRED
797 operation.

798 3.1.4 Character Set and Natural Language Operation Attributes

799 Some Job and Printer attributes have values that are text strings and names intended for human
800 understanding rather than machine understanding (see the 'text' and 'name' attribute syntax descriptions
801 in section 4.1). The following sections describe two special Operation Attributes called "attributes-
802 charset" and "attributes-natural-language". These attributes are always part of the Operation Attributes
803 group. For most attribute groups, the order of the attributes within the group is not important. However,
804 for these two attributes within the Operation Attributes group, the order is critical. The "attributes-
805 charset" attribute MUST be the first attribute in the group and the "attributes-natural-language" attribute
806 MUST be the second attribute in the group. In other words, these attributes MUST be supplied in every
807 IPP request and response, they MUST come first in the group, and MUST come in the specified order.
808 For job creation operations, the IPP Printer implementation saves these two attributes with the new Job
809 object as Job Description attributes. For the sake of brevity in this document, these operation attribute
810 descriptions are not repeated with every operation request and response, but have a reference back to this
811 section instead.

812 3.1.4.1 Request Operation Attributes

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

815 "attributes-charset" (charset):

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

821
822 All clients and IPP objects **MUST** support the 'utf-8' charset [RFC2279] and **MAY** support
823 additional charsets provided that they are registered with IANA [IANA-CS]. If the Printer object
824 does not support the client supplied charset value, the Printer object **MUST** reject the request, set
825 the "attributes-charset" to 'utf-8' in the response, and return the 'client-error-charset-not-
826 supported' status code and any 'text' or 'name' attributes using the 'utf-8' charset. The Printer
827 object **MUST** indicate the charset(s) supported as the values of the "charset-supported" Printer
828 attribute (see Section 4.4.15), so that the client can query to determine which charset(s) are
829 supported.

830
831 Note to client implementers: Since IPP objects are only required to support the 'utf-8' charset, in
832 order to maximize interoperability with multiple IPP object implementations, a client may want
833 to supply 'utf-8' in the "attributes-charset" operation attribute, even though the client is only
834 passing and able to present a simpler charset, such as US-ASCII or ISO-8859-1. Then the client
835 will have to filter out (or charset convert) those characters that are returned in the response that it
836 cannot present to its user. On the other hand, if both the client and the IPP objects also support a
837 charset in common besides utf-8, the client may want to use that charset in order to avoid charset
838 conversion or data loss.

839
840 See the 'charset' attribute syntax description in Section 4.1.7 for the syntax and semantic
841 interpretation of the values of this attribute and for example values.

842
843 "attributes-natural-language" (naturalLanguage):

844 This operation attribute identifies the natural language used by any 'text' and 'name' attributes that
845 the client is supplying in this request. This attribute also identifies the natural language that the
846 Printer object **SHOULD** use for all 'text' and 'name' attributes and status messages that the Printer
847 object returns in the response to this request.

848
849 There are no **REQUIRED** natural languages required for the Printer object to support. However,
850 the Printer object's "generated-natural-language-supported" attribute identifies the natural
851 languages supported by the Printer object and any contained Job objects for all text strings
852 generated by the IPP object. A client **MAY** query this attribute to determine which natural
853 language(s) are supported for generated messages.

854

855 For any of the attributes for which the Printer object generates text, i.e., for the "job-state-
856 message", "printer-state-message", and status messages (see Section 3.1.6), the Printer object
857 MUST be able to generate these text strings in any of its supported natural languages. If the
858 client requests a natural language that is not supported, the Printer object MUST return these
859 generated messages in the Printer's configured natural language as specified by the Printer's
860 "natural-language-configured" attribute" (see Section 4.4.16).

861
862 For other 'text' and 'name' attributes supplied by the client, authentication system, operator,
863 system administrator, or manufacturer (i.e., for "job-originating-user-name", "printer-name"
864 (name), "printer-location" (text), "printer-info" (text), and "printer-make-and-model" (text)), the
865 Printer object is only required to support the configured natural language of the Printer identified
866 by the Printer object's "natural-language-configured" attribute, though support of additional
867 natural languages for these attributes is permitted.

868
869 For any 'text' or 'name' attribute in the request that is in a different natural language than the value
870 supplied in the "attributes-natural-language" operation attribute, the client MUST use the Natural
871 Language Override mechanism (see sections 4.1.1.2 and 4.1.2.2) for each such attribute value
872 supplied. The client MAY use the Natural Language Override mechanism redundantly, i.e., use
873 it even when the value is in the same natural language as the value supplied in the "attributes-
874 natural-language" operation attribute of the request.

875
876 The IPP object MUST accept any natural language and any Natural Language Override, whether
877 the IPP object supports that natural language or not (and independent of the value of the "ipp-
878 attribute-fidelity" Operation attribute). That is the IPP object accepts all client supplied values no
879 matter what the values are in the Printer object's "generated-natural-language-supported"
880 attribute. That attribute, "generated-natural-language-supported", only applies to generated
881 messages, not client supplied messages. The IPP object MUST remember that natural language
882 for all client-supplied attributes, and when returning those attributes in response to a query, the
883 IPP object MUST indicate that natural language.

884
885 Each value whose attribute syntax type is 'text' or 'name' (see sections 4.1.1 and 4.1.2) has an
886 Associated Natural-Language. This document does not specify how this association is stored in a
887 Printer or Job object. When such a value is encoded in a request or response, the natural
888 language is either implicit or explicit:

- 889
- 890 • In the implicit case, the value contains only the text/name value, and the language is
891 specified by the "attributes-natural-language" operation attribute in the request or
892 response (see sections 4.1.1.1 textWithoutLanguage and 4.1.2.1
893 nameWithoutLanguage).
 - 894
 - 895 • In the explicit case (also known as the Natural-Language Override case), the value
896 contains both the language and the text/name value (see sections 4.1.1.2
897 textWithLanguage and 4.1.2.2 nameWithLanguage).
 - 898

899 For example, the "job-name" attribute MAY be supplied by the client in a create request. The
900 text value for this attribute will be in the natural language identified by the "attribute-natural-
901 language" attribute, or if different, as identified by the Natural Language Override mechanism. If
902 supplied, the IPP object will use the value of the "job-name" attribute to populate the Job object's
903 "job-name" attribute. Whenever any client queries the Job object's "job-name" attribute, the IPP
904 object returns the attribute as stored and uses the Natural Language Override mechanism to
905 specify the natural language, if it is different from that reported in the "attributes-natural-
906 language" operation attribute of the response. The IPP object MAY use the Natural Language
907 Override mechanism redundantly, i.e., use it even when the value is in the same natural language
908 as the value supplied in the "attributes-natural-language" operation attribute of the response.

909
910 An IPP object MUST NOT reject a request based on a supplied natural language in an
911 "attributes-natural-language" Operation attribute or in any attribute that uses the Natural
912 Language Override.

913
914 See the 'naturalLanguage' attribute syntax description in section 4.1.8 for the syntax and semantic
915 interpretation of the values of this attribute and for example values.

916
917 Clients SHOULD NOT supply 'text' or 'name' attributes that use an illegal combination of natural
918 language and charset. For example, suppose a Printer object supports charsets 'utf-8', 'iso-8859-1', and
919 'iso-8859-7'. Suppose also, that it supports natural languages 'en' (English), 'fr' (French), and 'el' (Greek).
920 Although the Printer object supports the charset 'iso-8859-1' and natural language 'el', it probably does
921 not support the combination of Greek text strings using the 'iso-8859-1' charset. The Printer object
922 handles this apparent incompatibility differently depending on the context in which it occurs:

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

931
932 In either case, the Printer object does not reject the request because of the apparent incompatibility. The
933 potential incompatible combination of charset and natural language can occur either at the global
934 operation level or at the Natural Language Override attribute-by-attribute level. In addition, since the
935 response always includes explicit charset and natural language information, there is never any question
936 or ambiguity in how the client interprets the response.

937 3.1.4.2 Response Operation Attributes

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

940 "attributes-charset" (charset):

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

946
947 If the Printer object supports more than just the 'utf-8' charset, the Printer object **MUST** be able to
948 code convert between each of the charsets supported on a highest fidelity possible basis in order
949 to return the 'text' and 'name' attributes in the charset requested by the client. However, some
950 information loss **MAY** occur during the charset conversion depending on the charsets involved.
951 For example, the Printer object may convert from a UTF-8 'a' to a US-ASCII 'a' (with no loss of
952 information), from an ISO Latin 1 CAPITAL LETTER A WITH ACUTE ACCENT to US-
953 ASCII 'A' (losing the accent), or from a UTF-8 Japanese Kanji character to some ISO Latin 1
954 error character indication such as '?', decimal code equivalent, or to the absence of a character,
955 depending on implementation.

956
957 Note: Whether an implementation that supports more than one charset stores the data in the
958 charset supplied by the client or code converts to one of the other supported charsets, depends on
959 implementation. The strategy should try to minimize loss of information during code conversion.
960 On each response, such an implementation converts from its internal charset to that requested.

961
962 "attributes-natural-language" (naturalLanguage):

963 This operation attribute identifies the natural language used by any 'text' and 'name' attributes that
964 the IPP object is returning in this response. Unlike the "attributes-charset" operation attribute,
965 the IPP object **NEED NOT** return the same value as that supplied by the client in the request.
966 The IPP object **MAY** return the natural language of the Job object or the Printer's configured
967 natural language as identified by the Printer object's "natural-language-configured" attribute,
968 rather than the natural language supplied by the client. For any 'text' or 'name' attribute or status
969 message in the response that is in a different natural language than the value returned in the
970 "attributes-natural-language" operation attribute, the IPP object **MUST** use the Natural Language
971 Override mechanism (see sections 4.1.1.2 and 4.1.2.2) on each attribute value returned. The IPP
972 object **MAY** use the Natural Language Override mechanism redundantly, i.e., use it even when
973 the value is in the same natural language as the value supplied in the "attributes-natural-
974 language" operation attribute of the response.

975 3.1.5 Operation Targets

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

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

- 982 - The Job object itself using the Job object's URI. In this case, the client identifies the target object
983 by supplying the correct URI in the "job-uri (uri)" operation attribute.
- 984 - The Printer object that created the Job object using both the Printer objects URI and the Job object's
985 Job ID. Since the Printer object that created the Job object generated the Job ID, it MUST be
986 able to correctly associate the client supplied Job ID with the correct Job object. The client
987 supplies the Printer object's URI in the "printer-uri (uri)" operation attribute and the Job object's
988 Job ID in the "job-id (integer(1:MAX))" operation attribute.

989

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

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

- 997 - In the case where there is only one operation target attribute (i.e., either only the "printer-uri"
998 attribute or only the "job-uri" attribute), that attribute MUST be the third attribute in the
999 operation attributes group.
- 1000 - In the case where Job operations use two operation target attributes (i.e., the "printer-uri" and "job-
1001 id" attributes), the "printer-uri" attribute MUST be the third attribute and the "job-id" attribute
1002 MUST be the fourth attribute.

1003

1004 In all cases, the target URIs contained within the body of IPP operation requests and responses must be
1005 in absolute format rather than relative format (a relative URL identifies a resource with the scope of the
1006 HTTP server, but does not include scheme, host or port).

1007 The following rules apply to the use of port numbers in URIs that identify IPP objects:

- 1008 1. If the URI scheme allows the port number to be explicitly included in the URI string, and a port
1009 number is specified within the URI, then that port number MUST be used by the client to contact
1010 the IPP object.
- 1011
- 1012 2. If the URI scheme allows the port number to be explicitly included in the URI string, and a port
1013 number is not specified within the URI, then default port number implied by that URI scheme
1014 MUST be used by the client to contact the IPP object.
- 1015
- 1016 3. If the URI scheme does not allow an explicit port number to be specified within the URI, then the
1017 default port number implied by that URI MUST be used by the client to contact the IPP object.

1018

1019 Note: The IPP "Encoding and Transport document [IPP-PRO] shows a mapping of IPP onto HTTP/1.1
1020 and defines a new default port number for using IPP over HTTP/1.1.

1021 3.1.6 Operation Status Codes and Messages

1022 Every operation response includes a REQUIRED "status-code" parameter and an OPTIONAL "status-
1023 message" operation attribute. The "status-code" provides information on the processing of a request. A
1024 "status-message" attribute provides a short textual description of the status of the operation. The status
1025 code is intended for use by automata, and the status message is intended for the human end user. If a
1026 response does include a "status-message" attribute, an IPP client NEED NOT examine or display the
1027 message, however it SHOULD do so in some implementation specific manner.

1028 The "status-code" value is a numeric value that has semantic meaning. The "status-code" syntax is
1029 similar to a "type2 enum" (see section 4.1 on "Attribute Syntaxes") except that values can range only
1030 from 0x0000 to 0x7FFF. Section 14 describes the status codes, assigns the numeric values, and suggests
1031 a corresponding status message for each status code. The "status-message" attribute's syntax is
1032 "text(255)". A client implementation of IPP SHOULD convert status code values into any localized
1033 message that has semantic meaning to the end user.

1034 If the Printer object supports the "status-message" operation attribute, the Printer object MUST be able
1035 to generate this message in any of the natural languages identified by the Printer object's "generated-
1036 natural-language-supported" attribute (see the "attributes-natural-language" operation attribute specified
1037 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
1038 generating this message, the Printer object uses the natural language indicated by the value of the
1039 "attributes-natural-language" in the client request if supported, otherwise the Printer object uses the
1040 value in the Printer object's own "natural-language-configured" attribute. If the Printer object supports
1041 the "status-message" operation attribute, it SHOULD use the REQUIRED 'utf-8' charset to return a status
1042 message for the following error status codes (see section 14): 'client-error-bad-request', 'client-error-
1043 charset-not-supported', 'server-error-internal-error', 'server-error-operation-not-supported', and 'server-
1044 error-version-not-supported'. In this case, it MUST set the value of the "attributes-charset" operation
1045 attribute to 'utf-8' in the error response.

1046 3.1.7 Versions

1047 Each operation request and response carries with it a "version-number" parameter. Each value of the
1048 "version-number" is in the form "X.Y" where X is the major version number and Y is the minor version
1049 number. By including a version number in the client request, it allows the client to identify which
1050 version of IPP it is interested in using. If the IPP object does not support that version, the object
1051 responds with a status code of 'server-error-version-not-supported' along with the closest version number
1052 that is supported (see section 14.1.5.4).

1053 There is no version negotiation per se. However, if after receiving a 'server-error-version-not-supported'
1054 status code from an IPP object, there is nothing that prevents a client from trying again with a different
1055 version number. In order to conform to IPP/1.1, an IPP object implementations MUST support versions
1056 '1.1' and 1.0.

1057 There is only one notion of "version number" that covers both IPP Model and IPP Protocol changes.
1058 Thus the version number MUST change when introducing a new version of the Model and Semantics
1059 document [IPP-MOD] or a new version of the "Encoding and Transport" document [IPP-PRO].

1060 Changes to the major version number indicate structural or syntactic changes that make it impossible for
1061 older version of IPP clients and Printer objects to correctly parse and process the new or changed
1062 attributes, operations and responses. If the major version number changes, the minor version numbers is
1063 set to zero. As an example, adding the "ipp-attribute-fidelity" attribute (if it had not been part of version
1064 '1.1'), would have required a change to the major version number. Items that might affect the changing
1065 of the major version number include any changes to the Model and Semantics document [IPP-MOD] or
1066 the "Encoding and Transport" document [IPP-PRO] itself, such as:

- 1067 - reordering of ordered attributes or attribute sets
- 1068 - changes to the syntax of existing attributes
- 1069 - changing Operation or Job Template attributes from OPTIONAL to REQUIRED and vice versa
- 1070 - adding REQUIRED (for an IPP object to support) operation attributes
- 1071 - adding REQUIRED (for an IPP object to support) operation attribute groups
- 1072 - adding values to existing operation attributes
- 1073 - adding REQUIRED operations

1074

1075 Changes to the minor version number indicate the addition of new features, attributes and attribute
1076 values that may not be understood by all IPP objects, but which can be ignored if not understood. Items
1077 that might affect the changing of the minor version number include any changes to the model objects and
1078 attributes but not the encoding and transport rules [IPP-PRO] (except adding attribute syntaxes).

1079 Examples of such changes are:

- 1080 - grouping all extensions not included in a previous version into a new version
- 1081 - adding new attribute values
- 1082 - adding new object attributes
- 1083 - adding OPTIONAL (for an IPP object to support) operation attributes (i.e., those attributes that an
1084 IPP object can ignore without confusing clients)
- 1085 - adding OPTIONAL (for an IPP object to support) operation attribute groups (i.e., those attributes
1086 that an IPP object can ignore without confusing clients)
- 1087 - adding new attribute syntaxes
- 1088 - adding OPTIONAL operations
- 1089 - changing Job Description attributes or Printer Description attributes from OPTIONAL to
1090 REQUIRED or vice versa.

1091

1092 The encoding of the "version-number" MUST NOT change over any version number (either major or
1093 minor). This rule guarantees that all future versions will be backwards compatible with all previous
1094 versions (at least for checking the "version-number"). In addition, any protocol elements (attributes,
1095 error codes, tags, etc.) that are not carried forward from one version to the next are deprecated so that
1096 they can never be reused with new semantics.

1097 Implementations that support a certain major version NEED NOT support ALL previous versions. As
1098 each new major version is defined (through the release of a new specification), that major version will
1099 specify which previous major versions MUST be supported in compliant implementations.

1100 3.1.8 Job Creation Operations

1101 In order to "submit a print job" and create a new Job object, a client issues a create request. A create
1102 request is any one of following three operation requests:

- 1103 - The Print-Job Request: A client that wants to submit a print job with only a single document uses
1104 the Print-Job operation. The operation allows for the client to "push" the document data to the
1105 Printer object by including the document data in the request itself.
1106
- 1107 - The Print-URI Request: A client that wants to submit a print job with only a single document
1108 (where the Printer object "pulls" the document data instead of the client "pushing" the data to the
1109 Printer object) uses the Print-URI operation. In this case, the client includes in the request only a
1110 URI reference to the document data (not the document data itself).
1111
- 1112 - The Create-Job Request: A client that wants to submit a print job with multiple documents uses the
1113 Create-Job operation. This operation is followed by an arbitrary number of Send-Document
1114 and/or Send-URI operations (each creating another document for the newly create Job object).
1115 The Send-Document operation includes the document data in the request (the client "pushes" the
1116 document data to the printer), and the Send-URI operation includes only a URI reference to the
1117 document data in the request (the Printer "pulls" the document data from the referenced location).
1118 The last Send-Document or Send-URI request for a given Job object includes a "last-document"
1119 operation attribute set to 'true' indicating that this is the last request.
1120

1121 Throughout this model specification, the term "create request" is used to refer to any of these three
1122 operation requests.

1123 A Create-Job operation followed by only one Send-Document operation is semantically equivalent to a
1124 Print-Job operation, however, for performance reasons, the client SHOULD use the Print-Job operation
1125 for all single document jobs. Also, Print-Job is a REQUIRED operation (all implementations MUST
1126 support it) whereas Create-Job is an OPTIONAL operation, hence some implementations might not
1127 support it.

1128 Job submission time is the point in time when a client issues a create request. The initial state of every
1129 Job object is the 'pending' or 'pending-held' state. Later, the Printer object begins processing the print
1130 job. At this point in time, the Job object's state moves to 'processing'. This is known as job processing
1131 time. There are validation checks that must be done at job submission time and others that must be
1132 performed at job processing time.

1133 At job submission time and at the time a Validate-Job operation is received, the Printer MUST do the
1134 following:

- 1135 1. Process the client supplied attributes and either accept or reject the request
- 1136 2. Validate the syntax of and support for the scheme of any client supplied URI

1137

1138 At job submission time the Printer object MUST validate whether or not the supplied attributes, attribute
1139 syntaxes, and values are supported by matching them with the Printer object's corresponding "xxx-

1140 supported" attributes. See section 3.2.1.2 for details. [IPP-IIG] presents suggested steps for an IPP
1141 object to either accept or reject any request and additional steps for processing create requests.

1142 At job submission time the Printer object NEED NOT perform the validation checks reserved for job
1143 processing time such as:

- 1144 1. Validating the document data
 - 1145 2. Validating the actual contents of any client supplied URI (resolve the reference and follow the link
1146 to the document data)
- 1147

1148 At job submission time, these additional job processing time validation checks are essentially useless,
1149 since they require actually parsing and interpreting the document data, are not guaranteed to be 100%
1150 accurate, and MUST be done, yet again, at job processing time. Also, in the case of a URI, checking for
1151 availability at job submission time does not guarantee availability at job processing time. In addition, at
1152 job processing time, the Printer object might discover any of the following conditions that were not
1153 detectable at job submission time:

- 1154 - runtime errors in the document data,
 - 1155 - nested document data that is in an unsupported format,
 - 1156 - the URI reference is no longer valid (i.e., the server hosting the document might be down), or
 - 1157 - any other job processing error
- 1158

1159 At job submission time, a Printer, especially a non-spooling Printer, MAY accept jobs that it does not
1160 have enough space for. In such a situation, a Printer MAY stop reading data from a client for an
1161 indefinite period of time. A client MUST be prepared for a write operation to block for an indefinite
1162 period of time (See section 5.1 on client conformance). When a printer has too little space for starting
1163 new jobs, it MAY reject a job with an error of 'server-error-busy'. When receiving such an error, a client
1164 MUST be prepared to keep submitting a job until the job submission succeeds. Issue 20

1165 At job processing time, since the Printer object has already responded with a successful status code in
1166 the response to the create request, if the Printer object detects an error, the Printer object is unable to
1167 inform the end user of the error with an operation status code. In this case, the Printer, depending on the
1168 error, can set the "job-state", "job-state-reasons", or "job-state-message" attributes to the appropriate
1169 value(s) so that later queries can report the correct job status.

1170 Note: Asynchronous notification of events is outside the scope of this IPP/1.1 document.

1171 3.2 Printer Operations

1172 All Printer operations are directed at Printer objects. A client MUST always supply the "printer-uri"
1173 operation attribute in order to identify the correct target of the operation.

1174 3.2.1 Print-Job Operation

1175 This REQUIRED operation allows a client to submit a print job with only one document and supply the
1176 document data (rather than just a reference to the data). See Section 16 for the suggested steps for
1177 processing create operations and their Operation and Job Template attributes.

1178 3.2.1.1 Print-Job Request

1179 The following groups of attributes are supplied as part of the Print-Job Request:

1180 Group 1: Operation Attributes

1181 Natural Language and Character Set:

1182 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1183 3.1.4.1. The Printer object MUST copy these values to the corresponding Job Description
1184 attributes described in sections 4.3.23 and 4.3.24.

1185

1186 Target:

1187 The "printer-uri" (uri) operation attribute which is the target for this operation as described in
1188 section 3.1.5.

1189

1190 Requesting User Name:

1191 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
1192 described in section 8.3.

1193

1194 "job-name" (name(MAX)):

1195 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1196 attribute. It contains the client supplied Job name. If this attribute is supplied by the client, its
1197 value is used for the "job-name" attribute of the newly created Job object. The client MAY
1198 automatically include any information that will help the end-user distinguish amongst his/her
1199 jobs, such as the name of the application program along with information from the document,
1200 such as the document name, document subject, or source file name. If this attribute is not
1201 supplied by the client, the Printer generates a name to use in the "job-name" attribute of the
1202 newly created Job object (see Section 4.3.5).

1203

1204 "ipp-attribute-fidelity" (boolean):

1205 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1206 attribute. The value 'true' indicates that total fidelity to client supplied Job Template attributes
1207 and values is required, else the Printer object MUST reject the Print-Job request. The value
1208 'false' indicates that a reasonable attempt to print the Job object is acceptable and the Printer
1209 object MUST accept the Print-job request. If not supplied, the Printer object assumes the value is
1210 'false'. All Printer objects MUST support both types of job processing. See section 16 for a full
1211 description of "ipp-attribute-fidelity" and its relationship to other attributes, especially the Printer
1212 object's "pdl-override-supported" attribute.

1213

1214 "document-name" (name(MAX)):

1215 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1216 attribute. It contains the client supplied document name. The document name MAY be
1217 different than the Job name. Typically, the client software automatically supplies the document
1218 name on behalf of the end user by using a file name or an application generated name. If this
1219 attribute is supplied, its value can be used in a manner defined by each implementation.
1220 Examples include: printed along with the Job (job start sheet, page adornments, etc.), used by
1221 accounting or resource tracking management tools, or even stored along with the document as a
1222 document level attribute. IPP/1.1 does not support the concept of document level attributes.

1223

1224 "document-format" (mimeMediaType) :

1225 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1226 attribute. The value of this attribute identifies the format of the supplied document data. If the
1227 client does not supply this attribute, the Printer object assumes that the document data is in the
1228 format defined by the Printer object's "document-format-default" attribute. If the client supplies
1229 this attribute, but the value is not supported by the Printer object, i.e., the value is not one of the
1230 values of the Printer object's "document-format-supported" attribute, the Printer object MUST
1231 copy the attribute and its value to the Unsupported Attributes response group, reject the request
1232 and return the 'client-error-document-format-not-supported' status code. Issue 11

1233

1234 "document-natural-language" (naturalLanguage):

1235 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports
1236 this attribute. This attribute specifies the natural language of the document for those document-
1237 formats that require a specification of the natural language in order to image the document
1238 unambiguously. There are no particular values required for the Printer object to support.

1239

1240 "compression" (type3 keyword)

1241 The client OPTIONALLY supplies this attribute. The Printer object ~~OPTIONALLY MUST~~
1242 ~~supports~~ this attribute and the "compression-supported" attribute (see section 4.4.29). The client
1243 supplied "compression" operation attribute identifies the compression algorithm used on the
1244 document data. If the client omits this attribute, the Printer object MUST assume that the data is
1245 not compressed. If the client supplies the attribute and the Printer object supports the attribute
1246 value, the Printer object uses the corresponding decompression algorithm on the document data.
1247 If the client supplies this attribute, but the value is not supported by the Printer object, i.e., the
1248 value is not one of the values of the Printer object's "compression-supported" attribute, the
1249 Printer object MUST copy the attribute and its value to the Unsupported Attributes response
1250 group, reject the request, and return the 'client-error-attributes-or-values-not-supported' status
1251 code. ~~If the client supplies this attribute, but this attribute is not supported by the Printer object,~~
1252 ~~i.e., the "compression-supported" attribute is not one of the Printer's Printer Description~~
1253 ~~attributes, the Printer object MUST copy the attribute to the Unsupported Attributes response~~
1254 ~~group changing the value to the out-of-band 'unsupported' value (see section 4.1), reject the~~
1255 ~~request, and return the 'client-error-attributes-or-values-not-supported' status code.~~ See section
1256 3.2.1.2 for returning unsupported attributes and values. Issue 28 Issue does this return client-
1257 error-compression-not-supported and does it have precedence over client-error-document-format-
1258 unsupported?

1259

1260 "job-k-octets" (integer(0:MAX))

1261 The client **OPTIONALLY** supplies this attribute. The Printer object **OPTIONALLY** supports
1262 this attribute and the "job-k-octets-supported" attribute (see section 4.4.30). The client supplied
1263 "job-k-octets" operation attribute identifies the total size of the document(s) in K octets being
1264 submitted (see section 4.3.17 for the complete semantics). If the client supplies the attribute and
1265 the Printer object supports the attribute, the value of the attribute is used to populate the Job
1266 object's "job-k-octets" Job Description attribute.

1267

1268 Note: For this attribute and the following two attributes ("job-impressions", and "job-media-
1269 sheets"), if the client supplies the attribute, but the Printer object does not support the attribute,
1270 the Printer object ignores the client-supplied value. If the client supplies the attribute and the
1271 Printer supports the attribute, and the value is within the range of the corresponding Printer
1272 object's "xxx-supported" attribute, the Printer object **MUST** use the value to populate the Job
1273 object's "xxx" attribute. If the client supplies the attribute and the Printer supports the attribute,
1274 but the value is outside the range of the corresponding Printer object's "xxx-supported" attribute,
1275 the Printer object **MUST** copy the attribute and its value to the Unsupported Attributes response
1276 group, reject the request, and return the 'client-error-attributes-or-values-not-supported' status
1277 code. If the client does not supply the attribute, the Printer object **MAY** choose to populate the
1278 corresponding Job object attribute depending on whether the Printer object supports the attribute
1279 and is able to calculate or discern the correct value.

1280

1281 "job-impressions" (integer(0:MAX))

1282 The client **OPTIONALLY** supplies this attribute. The Printer object **OPTIONALLY** supports
1283 this attribute and the "job-impressions-supported" attribute (see section 4.4.31). The client
1284 supplied "job-impressions" operation attribute identifies the total size in number of impressions
1285 of the document(s) being submitted (see section 4.3.18 for the complete semantics).

1286

1287 See note under "job-k-octets".

1288

1289 "job-media-sheets" (integer(0:MAX))

1290 The client **OPTIONALLY** supplies this attribute. The Printer object **OPTIONALLY** supports
1291 this attribute and the "job-media-sheets-supported" attribute (see section 4.4.32). The client
1292 supplied "job-media-sheets" operation attribute identifies the total number of media sheets to be
1293 produced for this job (see section 4.3.19 for the complete semantics).

1294

1295 See note under "job-k-octets".

1296

1297 Group 2: Job Template Attributes

1298 The client **OPTIONALLY** supplies a set of Job Template attributes as defined in section 4.2. If
1299 the client is not supplying any Job Template attributes in the request, the client **SHOULD** omit
1300 Group 2 rather than sending an empty group. However, a Printer object **MUST** be able to accept
1301 an empty group.

1302

1303 Group 3: Document Content

1304 The client **MUST** supply the document data to be processed.

1305

1306 Note: In addition to the **MANDATORY** parameters required for every operation request, the simplest
1307 Print-Job Request consists of just the "attributes-charset" and "attributes-natural-language" operation
1308 attributes; the "printer-uri" target operation attribute; the Document Content and nothing else. In this
1309 simple case, the Printer object:

- 1310 - creates a new Job object (the Job object contains a single document),
- 1311 - stores a generated Job name in the "job-name" attribute in the natural language and charset
1312 requested (see Section 3.1.4.1) (if those are supported, otherwise using the Printer object's default
1313 natural language and charset), and
- 1314 - at job processing time, uses its corresponding default value attributes for the supported Job
1315 Template attributes that were not supplied by the client as IPP attribute or embedded instructions
1316 in the document data.

1317

1318 3.2.1.2 Print-Job Response

1319 The Printer object **MUST** return to the client the following sets of attributes as part of the Print-Job
1320 Response:

1321 Group 1: Operation Attributes

1322 Status Message:

1323 In addition to the **REQUIRED** status code returned in every response, the response
1324 **OPTIONALLY** includes a "status-message" (text) operation attribute as described in sections 14
1325 and 3.1.6. If the client supplies unsupported or conflicting Job Template attributes or values, the
1326 Printer object **MUST** reject or accept the Print-Job request depending on the whether the client
1327 supplied a 'true' or 'false' value for the "ipp-attribute-fidelity" operation attribute. See the
1328 Implementer's Guide [IPP-IIG] for a complete description of the suggested steps for processing a
1329 create request.

1330

1331 Natural Language and Character Set:

1332 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1333 3.1.4.2.

1334

1335 Group 2: Unsupported Attributes

1336 This is a set of **all** Operation and Job Template attributes supplied by the client (in the request)
1337 that are not supported by the Printer object or that conflict with one another (see the
1338 Implementer's Guide [IPP-IIG]). If the Printer object is not returning any Unsupported Attributes
1339 in the response, the Printer object **SHOULD** omit Group 2 rather than sending an empty group.
1340 However, a client **MUST** be able to accept an empty group. **Issue 18**

1341

1342 The unsupported attributes MUST contain ALL of those attributes and ONLY those attributes
1343 that meet the following two conditions: a) they were supplied in the request and b) they are
1344 unsupported by the printer, where "unsupported" is defined below. Issues 18, 23, and 27
1345

1346 Unsupported attributes fall into three categories:

- 1347 1. The Printer object does not support the supplied attribute (no matter what the attribute syntax
1348 or value).
- 1349 2. The Printer object does support the attribute, but does not support some or all of the particular
1350 attribute syntaxes or values supplied by the client (i.e., the Printer object does not have
1351 those attribute syntaxes or values in its corresponding "xxx-supported" attribute).
- 1352 3. The Printer object does support the attributes and values supplied, but the particular values are
1353 in conflict with one another, because they violate a constraint, such as not being able to
1354 staple transparencies.

1355
1356
1357 In the case of an unsupported attribute name, the Printer object returns the client-supplied
1358 attribute with a substituted "~~out-of-band~~"-value of 'unsupported'. This value's syntax type is "out-
1359 of-band" and it's encoding is defined by special rules for "out-of-band" values in the "Encoding
1360 and Transport" specification [IPP-PRO]. It's value ~~indicating~~ indicates no support for the
1361 attribute itself (see the beginning of section 4.1). Issue 12
1362

1363 In the case of a supported attribute with one or more unsupported attribute syntaxes or values, the
1364 Printer object simply returns the client-supplied attribute with the unsupported attribute syntaxes
1365 or values as supplied by the client. This indicates support for the attribute, but no support for that
1366 particular attribute syntax or value. If the client supplies a multi-valued attribute with more than
1367 one value and the Printer object supports the attribute but only supports a subset of the client-
1368 supplied attribute syntaxes or values, the Printer object MUST return only those attribute
1369 syntaxes or values that are unsupported.
1370

1371 In the case of two (or more) supported attribute values that are in conflict with one another
1372 (although each is supported independently, the values conflict when requested together within the
1373 same job), the Printer object MUST return all the values that it ignores or substitutes to resolve
1374 the conflict, but not any of the values that it is still using. The choice for exactly how to resolve
1375 the conflict is implementation dependent. See The Implementer's Guide [IPP-IIG] for an
1376 example.
1377

1378 In these three cases, the value of the "ipp-attribute-fidelity" supplied by the client does not affect
1379 what the Printer object returns. The value of "ipp-attribute-fidelity" only affects whether the
1380 Print-Job operation is accepted or rejected. If the job is accepted, the client may query the job
1381 using the Get-Job-Attributes operation requesting the unsupported attributes that were returned in
1382 the create response to see which attributes were ignored (not stored on the Job object) and which
1383 attributes were stored with other (substituted) values.
1384

1385 Group 3: Job Object Attributes

1386 "job-uri" (uri):
1387 The Printer object MUST return the Job object's URI by returning the contents of the
1388 REQUIRED "job-uri" Job object attribute. The client uses the Job object's URI when directing
1389 operations at the Job object. The Printer object always uses its configured security policy when
1390 creating the new URI. However, if the Printer object supports more than one URI, the Printer
1391 object also uses information about which URI was used in the Print-Job Request to generated the
1392 new URI so that the new URI references the correct access channel. In other words, if the Print-
1393 Job Request comes in over a secure channel, the Printer object MUST generate a Job URI that
1394 uses the secure channel as well.
1395

1396 "job-id" (integer(1:MAX)):
1397 The Printer object MUST return the Job object's Job ID by returning the REQUIRED "job-id"
1398 Job object attribute. The client uses this "job-id" attribute in conjunction with the "printer-uri"
1399 attribute used in the Print-Job Request when directing Job operations at the Printer object.
1400

1401 "job-state":
1402 The Printer object MUST return the Job object's REQUIRED "job-state" attribute. The value of
1403 this attribute (along with the value of the next attribute "job-state-reasons") is taken from a
1404 "snapshot" of the new Job object at some meaningful point in time (implementation defined)
1405 between when the Printer object receives the Print-Job Request and when the Printer object
1406 returns the response.
1407

1408 "job-state-reasons":
1409 The Printer object OPTIONALLY returns the Job object's OPTIONAL "job-state-reasons"
1410 attribute. If the Printer object supports this attribute then it MUST be returned in the response. If
1411 this attribute is not returned in the response, the client can assume that the "job-state-reasons"
1412 attribute is not supported and will not be returned in a subsequent Job object query.
1413

1414 "job-state-message":
1415 The Printer object OPTIONALLY returns the Job object's OPTIONAL "job-state-message"
1416 attribute. If the Printer object supports this attribute then it MUST be returned in the response. If
1417 this attribute is not returned in the response, the client can assume that the "job-state-message"
1418 attribute is not supported and will not be returned in a subsequent Job object query.
1419

1420 "number-of-intervening-jobs":
1421 The Printer object OPTIONALLY returns the Job object's OPTIONAL "number-of-intervening-
1422 jobs" attribute. If the Printer object supports this attribute then it MUST be returned in the
1423 response. If this attribute is not returned in the response, the client can assume that the "number-
1424 of-intervening-jobs" attribute is not supported and will not be returned in a subsequent Job object
1425 query.
1426

1427 Note: Since any printer state information which affects a job's state is reflected in the "job-state"
1428 and "job-state-reasons" attributes, it is sufficient to return only these attributes and no specific
1429 printer status attributes.
1430

1431 Note: In addition to the MANDATORY parameters required for every operation response, the simplest
1432 response consists of the just the "attributes-charset" and "attributes-natural-language" operation
1433 attributes and the "job-uri", "job-id", and "job-state" Job Object Attributes. In this simplest case, the
1434 status code is "successful-ok" and there is no "status-message" operation attribute.

1435 3.2.2 Print-URI Operation

1436 This OPTIONAL operation is identical to the Print-Job operation (section 3.2.1) except that a client
1437 supplies a URI reference to the document data using the "document-uri" (uri) operation attribute (in
1438 Group 1) rather than including the document data itself. Before returning the response, the Printer
1439 MUST validate that the Printer supports the retrieval method (e.g., http, ftp, etc.) implied by the URI,
1440 and MUST check for valid URI syntax. If the client-supplied URI scheme is not supported, i.e. the value
1441 is not in the Printer object's "referenced-uri-scheme-supported" attribute, the Printer object MUST reject
1442 the request and return the 'client-error-uri-scheme-not-supported' status code. See The Implementer's
1443 Guide [IPP-IIG] for suggested additional checks. The Printer NEED NOT follow the reference and
1444 validate the contents of the reference.

1445 If the Printer object supports this operation, it MUST support the "reference-uri-schemes-supported"
1446 Printer attribute (see section 4.4.24).

1447 It is up to the IPP object to interpret the URI and subsequently "pull" the document from the source
1448 referenced by the URI string.

1449 3.2.3 Validate-Job Operation

1450 This REQUIRED operation is similar to the Print-Job operation (section 3.2.1) except that a client
1451 supplies no document data and the Printer allocates no resources (i.e., it does not create a new Job
1452 object). This operation is used only to verify capabilities of a printer object against whatever attributes
1453 are supplied by the client in the Validate-Job request. By using the Validate-Job operation a client can
1454 validate that an identical Print-Job operation (with the document data) would be accepted. The Validate-
1455 Job operation also performs the same security negotiation as the Print-Job operation (see section 8), so
1456 that a client can check that the client and Printer object security requirements can be met before
1457 performing a Print-Job operation.

1458 Note: The Validate-Job operation does not accept a "document-uri" attribute in order to allow a client to
1459 check that the same Print-URI operation will be accepted, since the client doesn't send the data with the
1460 Print-URI operation. The client SHOULD just issue the Print-URI request.

1461 The Printer object returns the same status codes, Operation Attributes (Group 1) and Unsupported
1462 Attributes (Group 2) as the Print-Job operation. However, no Job Object Attributes (Group 3) are
1463 returned, since no Job object is created.

1464 3.2.4 Create-Job Operation

1465 This OPTIONAL operation is similar to the Print-Job operation (section 3.2.1) except that in the Create-
1466 Job request, a client does not supply document data or any reference to document data. Also, the client

1467 does not supply any of the "document-name", "document-format", "compression", or "document-natural-
1468 language" operation attributes. This operation is followed by one or more Send-Document or Send-URI
1469 operations. In each of those operation requests, the client OPTIONALLY supplies the "document-
1470 name", "document-format", and "document-natural-language" attributes for each document in the multi-
1471 document Job object.

1472 If a Printer object supports the Create-Job operation, it MUST also support the Send-Document
1473 operation and also MAY support the Send-URI operation.

1474 If the Printer object supports this operation, it MUST support the "multiple-operation-time-out" Printer
1475 attribute (see section 4.4.28).

1476 Note: after the Create-Job operation has completed, the value of the "job-state" attribute is similar to the
1477 "job-state" after a Print-Job, even though no document-data has arrived. A non-spooling printer that
1478 doesn't implement the 'pending' job state may even set the "job-state" to 'processing'. The 'job-data-
1479 insufficient' job-state-reason indicates that marking cannot begin until sufficient data has arrived. Issue
1480 13

1481 3.2.5 Get-Printer-Attributes Operation

1482 This REQUIRED operation allows a client to request the values of the attributes of a Printer object. In
1483 the request, the client supplies the set of Printer attribute names and/or attribute group names in which
1484 the requester is interested. In the response, the Printer object returns a corresponding attribute set with
1485 the appropriate attribute values filled in.

1486 For Printer objects, the possible names of attribute groups are:

- 1487 - 'job-template': all of the Job Template attributes that apply to a Printer object (the last two columns
1488 of the table in Section 4.2).
 - 1489 - 'printer-description': the attributes specified in Section 4.4.
 - 1490 - 'all': the special group 'all' that includes all supported attributes.
- 1491

1492 Since a client MAY request specific attributes or named groups, there is a potential that there is some
1493 overlap. For example, if a client requests, 'printer-name' and 'all', the client is actually requesting the
1494 "printer-name" attribute twice: once by naming it explicitly, and once by inclusion in the 'all' group. In
1495 such cases, the Printer object NEED NOT return each attribute only once in the response even if it is
1496 requested multiple times. The client SHOULD NOT request the same attribute in multiple ways.

1497 It is NOT REQUIRED that a Printer object support all attributes belonging to a group (since some
1498 attributes are OPTIONAL). However, it is REQUIRED that each Printer object support all group names.

1499 3.2.5.1 Get-Printer-Attributes Request

1500 The following sets of attributes are part of the Get-Printer-Attributes Request:

1501 Group 1: Operation Attributes

1502 Natural Language and Character Set:

1503 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1504 3.1.4.1.

1505
1506 Target:

1507 The "printer-uri" (uri) operation attribute which is the target for this operation as described in
1508 section 3.1.5.

1509
1510 Requesting User Name:

1511 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
1512 described in section 8.3.

1513
1514 "requested-attributes" (1setOf keyword) :

1515 The client OPTIONALLY supplies a set of attribute names and/or attribute group names in
1516 whose values the requester is interested. The Printer object MUST support this attribute. If the
1517 client omits this attribute, the Printer MUST respond as if this attribute had been supplied with a
1518 value of 'all'.

1519
1520 "document-format" (mimeMediaType) :

1521 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1522 attribute. This attribute is useful for a Printer object to determine the set of supported attribute
1523 values that relate to the requested document format. The Printer object MUST return the
1524 attributes and values that it uses to validate a job on a create or Validate-Job operation in which
1525 this document format is supplied. The Printer object SHOULD return only (1) those attributes
1526 that are supported for the specified format and (2) the attribute values that are supported for the
1527 specified document format. By specifying the document format, the client can get the Printer
1528 object to eliminate the attributes and values that are not supported for a specific document
1529 format. For example, a Printer object might have multiple interpreters to support both
1530 'application/postscript' (for PostScript) and 'text/plain' (for text) documents. However, for only
1531 one of those interpreters might the Printer object be able to support "number-up" with values of
1532 '1', '2', and '4'. For the other interpreter it might be able to only support "number-up" with a value
1533 of '1'. Thus a client can use the Get-Printer-Attributes operation to obtain the attributes and
1534 values that will be used to accept/reject a create job operation.

1535
1536 If the Printer object does not distinguish between different sets of supported values for each
1537 different document format when validating jobs in the create and Validate-Job operations, it
1538 MUST NOT distinguish between different document formats in the Get-Printer-Attributes
1539 operation. If the Printer object does distinguish between different sets of supported values for
1540 each different document format specified by the client, this specialization applies only to the
1541 following Printer object attributes:

- 1542
- 1543 - Printer attributes that are Job Template attributes ("xxx-default" "xxx-supported", and "xxx-
1544 ready" in the Table in Section 4.2),
 - 1545 - "pdl-override-supported",
 - 1546 - "compression-supported",

- 1547 - "job-k-octets-supported",
- 1548 - "job-impressions-supported",
- 1549 - "job-media-sheets-supported"
- 1550 - "printer-driver-installer",
- 1551 - "color-supported", and
- 1552 - "reference-uri-schemes-supported"

1553

1554 The values of all other Printer object attributes (including "document-format-supported") remain
1555 invariant with respect to the client supplied document format (except for new Printer description
1556 attribute as registered according to section 6.2).

1557

1558 If the client omits this "document-format" operation attribute, the Printer object MUST respond
1559 as if the attribute had been supplied with the value of the Printer object's "document-format-
1560 default" attribute. It is recommended that the client always supply a value for "document-
1561 format", since the Printer object's "document-format-default" may be 'application/octet-stream',
1562 in which case the returned attributes and values are for the union of the document formats that
1563 the Printer can automatically sense. For more details, see the description of the
1564 'mimeType' attribute syntax in section 4.1.9.

1565

1566 If the client supplies a value for the "document-format" Operation attribute that is not supported
1567 by the Printer, i.e., is not among the values of the Printer object's "document-format-supported"
1568 attribute, the Printer object MUST reject the operation and return the 'client-error-document-
1569 format-not-supported' status code.

1570

1571 3.2.5.2 Get-Printer-Attributes Response

1572 The Printer object returns the following sets of attributes as part of the Get-Printer-Attributes Response:

1573 Group 1: Operation Attributes

1574 Status Message:

1575 In addition to the REQUIRED status code returned in every response, the response
1576 OPTIONALLY includes a "status-message" (text) operation attribute as described in section
1577 3.1.6.

1578

1579 Natural Language and Character Set:

1580 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1581 3.1.4.2.

1582

1583 Group 2: Unsupported Attributes

1584 This is a set of Operation attributes supplied by the client (in the request) that are not supported
1585 by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16). The response
1586 NEED NOT contain the "requested-attributes" operation attribute with any supplied values
1587 (attribute keywords) that were requested by the client but are not supported by the IPP object. **If**

1588 the Printer does include unsupported attributes referenced in "requested-attributes" and such
1589 attributes include group names, such as "all", the unsupported attributes MUST NOT include
1590 attributes described in the standard but not supported by the implementation. [Issue 23](#) If the
1591 Printer object is not returning any Unsupported Attributes in the response, the Printer object
1592 SHOULD omit Group 2 rather than sending an empty group. However, a client MUST be able
1593 to accept an empty group.
1594

1595 Group 3: Printer Object Attributes

1596 This is the set of requested attributes and their current values. The Printer object ignores (does
1597 not respond with) any requested attribute which is not supported. The Printer object MAY
1598 respond with a subset of the supported attributes and values, depending on the security policy in
1599 force. However, the Printer object MUST respond with the 'unknown' value for any supported
1600 attribute (including all REQUIRED attributes) for which the Printer object does not know the
1601 value. Also the Printer object MUST respond with the 'no-value' for any supported attribute
1602 (including all REQUIRED attributes) for which the system administrator has not configured a
1603 value. See the description of the "out-of-band" values in the beginning of Section 4.1.
1604

1605 3.2.6 Get-Jobs Operation

1606 This REQUIRED operation allows a client to retrieve the list of Job objects belonging to the target
1607 Printer object. The client may also supply a list of Job attribute names and/or attribute group names. A
1608 group of Job object attributes will be returned for each Job object that is returned.

1609 This operation is similar to the Get-Job-Attributes operation, except that this Get-Jobs operation returns
1610 attributes from possibly more than one object (see the description of Job attribute group names in section
1611 3.3.4).

1612 3.2.6.1 Get-Jobs Request

1613 The client submits the Get-Jobs request to a Printer object.

1614 The following groups of attributes are part of the Get-Jobs Request:

1615 Group 1: Operation Attributes

1616 Natural Language and Character Set:

1617 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1618 3.1.4.1.

1619
1620 Target:

1621 The "printer-uri" (uri) operation attribute which is the target for this operation as described in
1622 section 3.1.5.
1623

1624 Requesting User Name:

1625 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
1626 described in section 8.3.

1627

1628 "limit" (integer(1:MAX)):

1629 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1630 attribute. It is an integer value that ~~indicates a limit to the number of Job objects returned.~~
1631 determines the maximum number of jobs that a client will receive from the Printer even if
1632 "which-jobs" or "my-jobs" restrict which jobs are returned. -The limit is a "stateless limit" in that
1633 if the value supplied by the client is 'N', then only the first 'N' jobs are returned in the Get-Jobs
1634 Response. There is no mechanism to allow for the next 'M' jobs after the first 'N' jobs. If the
1635 client does not supply this attribute, the Printer object responds with all applicable jobs. **Issue 8**
1636

1637

1638 "requested-attributes" (1setOf keyword):

1639 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1640 attribute. It is a set of Job attribute names and/or attribute groups names in whose values the
1641 requester is interested. This set of attributes is returned for each Job object that is returned. The
1642 allowed attribute group names are the same as those defined in the Get-Job-Attributes operation
1643 in section 3.3.4. If the client does not supply this attribute, the Printer MUST respond as if the
1644 client had supplied this attribute with two values: 'job-uri' and 'job-id'.

1645

1646 "which-jobs" (keyword):

1647 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1648 attribute. It indicates which Job objects MUST be returned by the Printer object. The values for
1649 this attribute are:

1650

1650 'completed': This includes any Job object whose state is 'completed', 'canceled', or 'aborted'.

1651

1651 'not-completed': This includes any Job object whose state is 'pending', 'processing',

1652

1652 'processing-stopped', or 'pending-held'.

1653

1654 A Printer object MUST support both values. However, if the implementation does not keep jobs
1655 in the 'completed', 'canceled', and 'aborted' states, then it returns no jobs when the 'completed'
1656 value is supplied.

1657

1658 If a client supplies some other value, the Printer object MUST copy the attribute and the
1659 unsupported value to the Unsupported Attributes response group, reject the request, and return
1660 the 'client-error-attributes-or-values-not-supported' status code.

1661

1662 If the client does not supply this attribute, the Printer object MUST respond as if the client had
1663 supplied the attribute with a value of 'not-completed'.

1664

1665 "my-jobs" (boolean):

1666 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1667 attribute. It indicates whether all jobs or just the jobs submitted by the requesting user of this
1668 request MUST be returned by the Printer object. If the client does not supply this attribute, the

1669 Printer object MUST respond as if the client had supplied the attribute with a value of 'false', i.e.,
1670 all jobs. The means for authenticating the requesting user and matching the jobs is described in
1671 section 8.

1672 3.2.6.2 Get-Jobs Response

1673 The Printer object returns all of the Job objects that match the criteria as defined by the attribute values
1674 supplied by the client in the request. It is possible that no Job objects are returned since there may
1675 literally be no Job objects at the Printer, or there may be no Job objects that match the criteria supplied
1676 by the client. If the client requests any Job attributes at all, there is a set of Job Object Attributes
1677 returned for each Job object.

1678 It is not an error for the Printer to return 0 jobs. If the response returns 0 jobs because there are no jobs
1679 matching the criteria, and the request would have returned 1 or more jobs with a status code of
1680 'successful-ok' if there had been jobs matching the criteria, then the status code for 0 jobs MUST be
1681 'successful-ok'. Issue 24

1682 Group 1: Operation Attributes

1683 Status Message:

1684 In addition to the REQUIRED status code returned in every response, the response
1685 OPTIONALLY includes a "status-message" (text) operation attribute as described in sections 14
1686 and 3.1.6.

1687 Natural Language and Character Set:

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

1692 Group 2: Unsupported Attributes

1693 This is a set of Operation attributes supplied by the client (in the request) that are not supported
1694 by the Printer object or that conflict with one another (see sections 3.2.1.2 and the Implementer's
1695 Guide [IPP-IIG]). The response NEED NOT contain the "requested-attributes" operation
1696 attribute with any supplied values (attribute keywords) that were requested by the client but are
1697 not supported by the IPP object. If the Printer does include unsupported attributes referenced in
1698 "requested-attributes" and such attributes include group names, such as "all", the unsupported
1699 attributes MUST NOT include attributes described in the standard but not supported by the
1700 implementation. Issue 23 If the Printer object is not returning any Unsupported Attributes in
1701 the response, the Printer object SHOULD omit Group 2 rather than sending an empty group.
1702 However, a client MUST be able to accept an empty group.

1704 Groups 3 to N: Job Object Attributes

1705 The Printer object responds with one set of Job Object Attributes for each returned Job object.
1706 The Printer object ignores (does not respond with) any requested attribute or value which is not
1707 supported or which is restricted by the security policy in force, including whether the requesting
1708 user is the user that submitted the job (job originating user) or not (see section 8). However, the

1709 Printer object MUST respond with the 'unknown' value for any supported attribute (including all
1710 REQUIRED attributes) for which the Printer object does not know the value, unless it would
1711 violate the security policy. See the description of the "out-of-band" values in the beginning of
1712 Section 4.1.

1713

1714 Jobs are returned in the following order:

- 1715 - If the client requests all 'completed' Jobs (Jobs in the 'completed', 'aborted', or 'canceled'
1716 states), then the Jobs are returned newest to oldest (with respect to actual completion
1717 time)
- 1718 - If the client requests all 'not-completed' Jobs (Jobs in the 'pending', 'processing', 'pending-
1719 held', and 'processing-stopped' states), then Jobs are returned in relative chronological
1720 order of expected time to complete (based on whatever scheduling algorithm is
1721 configured for the Printer object).

1722

1723 3.2.7 Pause-Printer Operation

1724 This OPTIONAL operation allows a client to stop the Printer object from scheduling jobs on all its
1725 devices. Depending on implementation, the Pause-Printer operation MAY also stop the Printer from
1726 processing the current job or jobs. Any job that is currently being printed is either stopped as soon as the
1727 implementation permits or is completed, depending on implementation. The Printer object MUST still
1728 accept create operations to create new jobs, but MUST prevent any jobs from entering the 'processing'
1729 state.

1730 If the Pause-Printer operation is supported, then the Resume-Printer operation MUST be supported, and
1731 vice-versa.

1732 The IPP Printer stops the current job(s) on its device(s) that were in the 'processing' or 'processing-
1733 stopped' states as soon as the implementation permits. If the implementation supports the "printer-state-
1734 reasons" attribute and the devices will take appreciable time to stop, the IPP Printer adds the 'moving-to-
1735 paused' value to the Printer object's "printer-state-reasons" attribute (see section 4.4.11). When the
1736 device(s) have all stopped, the IPP Printer transitions the Printer object to the 'stopped' state, removes the
1737 'moving-to-paused' value, if present, and adds the 'paused' value to the Printer object's "printer-state-
1738 reasons" attribute.

1739 When the current job(s) complete that were in the 'processing' state, the IPP Printer transitions them to
1740 the 'completed' state. When the current job(s) stop in mid processing that were in the 'processing' state,
1741 the IPP Printer transitions them to the 'processing-stopped' state and, if the "job-state-reasons" attribute is
1742 supported, adds the 'printer-stopped' value to the job's "job-state-reasons" attribute.

1743 Note: for any jobs that are 'pending' or 'pending-held', the 'printer-stopped' value of the jobs' "job-state-
1744 reasons" attribute also applies. However, the IPP Printer NEED NOT update those jobs' "job-state-
1745 reasons" attributes and only need return the 'printer-stopped' value when those jobs are queried (so-called
1746 "lazy evaluation").

1747 Whether the Pause-Printer operation affects jobs that were submitted to the device from other sources
 1748 than the IPP Printer object in the same way that the Pause-Printer operation affects jobs that were
 1749 submitted to the IPP Printer object using IPP, depends on implementation, i.e., on whether the IPP
 1750 protocol is being used as a universal management protocol or just to manage IPP jobs, respectively.

1751 The IPP Printer MUST accept the request in any state and transition the Printer to the indicated new
 1752 "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'

1753 *Access Rights:* The requesting user must be an operator or administrator of the Printer object.
 1754 Otherwise, the IPP Printer MUST reject the operation and return: 'client-error-forbidden', 'client-error-
 1755 not-authenticated', or 'client-error-not-authorized' as appropriate.

1756 3.2.7.1 Pause-Printer Request

1757 The following groups of attributes are part of the Pause-Printer Request:

1758 Group 1: Operation Attributes

1759 Natural Language and Character Set:

1760 The "attributes-charset" and "attributes-natural-language" attributes as described in section
 1761 3.1.4.1.

1762

1763 Target:

1764 The "printer-uri" (uri) operation attribute which is the target for this operation as described in
 1765 section 3.1.5.

1766

1767 Requesting User Name:

1768 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
 1769 described in section 8.3.

1770 3.2.7.2 Pause-Printer Response

1771 The following groups of attributes are part of the Pause-Printer Response:

1772 Group 1: Operation Attributes

1773 Status Message:

1774 In addition to the REQUIRED status code returned in every response, the response
1775 OPTIONALLY includes a "status-message" (text) operation attribute as described in section
1776 3.1.6.

1777
1778 Natural Language and Character Set:

1779 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1780 3.1.4.2.

1781

1782 Group 2: Unsupported Attributes

1783 This is a set of Operation attributes supplied by the client (in the request) that are not supported
1784 by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16).

1785 3.2.8 Resume-Printer Operation

1786 This operation allows a client to resume the Printer object scheduling jobs on all its devices. If the
1787 Printer object supports the "printer-state-reasons" attribute, it MUST remove the 'paused' and 'moving-
1788 to-paused' values from the Printer object's "printer-state-reasons" attribute, if present. If there are no
1789 other reasons to keep a device paused (such as media-jam), the IPP Printer transitions itself to the
1790 'processing' or 'idle' states, depending on whether there are jobs to be processed or not, respectively, and
1791 the device(s) resume processing jobs.

1792 If the Pause-Printer operation is supported, then the Resume-Printer operation MUST be supported, and
1793 vice-versa.

1794 The IPP Printer removes the 'printer-stopped' value from any job's "job-state-reasons" attributes
1795 contained in that Printer.

1796 The IPP Printer MUST accept the request in any state, transition the Printer object to the indicated new
1797 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'
'stopped'	'processing'	'successful-ok'; when there are jobs to be processed
'stopped'	'idle'	'successful-ok'; when there are no jobs to be processed.

1798 *Access Rights:* The requesting user must be an operator or administrator of the Printer object.
1799 Otherwise, the IPP Printer MUST reject the operation and return: 'client-error-forbidden', 'client-error-
1800 not-authenticated', or 'client-error-not-authorized' as appropriate.

1801 The Resume-Printer Request and Resume-Printer Response have the same attribute groups and attributes
1802 as the Pause-Printer operation (see sections 3.2.7.1 and 3.2.7.2).

1803 3.2.9 Purge-Jobs Operation

1804 This OPTIONAL operation allows a client to remove all jobs from an IPP Printer object, regardless of
1805 their job states, including jobs in the Printer object's Job History (see Section 4.3.7.1). After a Purge-
1806 Jobs operation has been performed, a Printer object MUST return no jobs in subsequent Get-Job-
1807 Attributes and Get-Jobs responses (until new jobs are submitted).

1808 Whether the Purge-Jobs (and Get-Jobs) operation affects jobs that were submitted to the device from
1809 other sources than the IPP Printer object in the same way that the Purge-Jobs operation affects jobs that
1810 were submitted to the IPP Printer object using IPP, depends on implementation, i.e., on whether the IPP
1811 protocol is being used as a universal management protocol or just to manage IPP jobs, respectively.

1812 Note: if an operator wants to cancel all jobs without clearing out the Job History, the operator uses the
1813 Cancel-Job operation on each job instead of using the Purge-Job operation.

1814 The Printer object MUST accept this operation in any state and transition the Printer object to the 'idle'
1815 state.

1816 *Access Rights:* The requesting user must be an operator or administrator of the Printer object.
1817 Otherwise, the IPP object MUST reject the operation and return: client-error-forbidden, client-error-not-
1818 authenticated, and client-error-not-authorized as appropriate.

1819 The Purge-Jobs Request and Purge-Jobs Response have the same attribute groups and attributes as the
1820 Pause-Printer operation (see sections 3.2.7.1 and 3.2.7.2).

1821 3.3 Job Operations

1822 All Job operations are directed at Job objects. A client MUST always supply some means of identifying
1823 the Job object in order to identify the correct target of the operation. That job identification MAY either
1824 be a single Job URI or a combination of a Printer URI with a Job ID. The IPP object implementation
1825 MUST support both forms of identification for every job.

1826 3.3.1 Send-Document Operation

1827 This OPTIONAL operation allows a client to create a multi-document Job object that is initially "empty"
1828 (contains no documents). In the Create-Job response, the Printer object returns the Job object's URI (the
1829 "job-uri" attribute) and the Job object's 32-bit identifier (the "job-id" attribute). For each new document
1830 that the client desires to add, the client uses a Send-Document operation. Each Send-Document Request
1831 contains the entire stream of document data for one document.

1832 The authenticated user of the Send-Document operation MUST be the same as the authenticated user of
1833 the Create-Job operation whose job-id or job-uri is the target of the Send-Document operation. Issue 19

1834 Since the Create-Job and the send operations (Send-Document or Send-URI operations) that follow
1835 could occur over an arbitrarily long period of time for a particular job, a client MUST send another send
1836 operation within an IPP Printer defined minimum time interval after the receipt of the previous request

1837 for the job. If a Printer object supports multiple document jobs, the Printer object MUST support the
1838 "multiple-operation-time-out" attribute (see section 4.4.28). This attribute indicates the minimum
1839 number of seconds the Printer object will wait for the next send operation before taking some recovery
1840 action.

1841 An IPP object MUST recover from an errant client that does not supply a send operation, sometime after
1842 the minimum time interval specified by the Printer object's "multiple-operation-time-out" attribute. Such
1843 recovery MAY include any of the following or other recovery actions:

- 1844 1. Assume that the Job is an invalid job, start the process of changing the job state to 'aborted', add
1845 the 'aborted-by-system' value to the job's "job-state-reasons" attribute (see section 4.3.8), if
1846 supported, and clean up all resources associated with the Job. In this case, if another send
1847 operation is finally received, the Printer responds with an "client-error-not-possible" or "client-
1848 error-not-found" depending on whether or not the Job object is still around when the send
1849 operation finally arrives.
- 1850 2. Assume that the last send operation received was in fact the last document (as if the "last-
1851 document" flag had been set to 'true'), close the Job object, and proceed to process it (i.e., move
1852 the Job's state to 'pending').
- 1853 3. Assume that the last send operation received was in fact the last document, close the Job, but
1854 move it to the 'pending-held' and add the 'submission-interrupted' value to the job's "job-state-
1855 reasons" attribute (see section 4.3.8), if supported. This action allows the user or an operator to
1856 determine whether to continue processing the Job by moving it back to the 'pending' state using
1857 the Release-Job operation (see section 3.3.6) or to cancel the job using the Cancel-Job operation
1858 (see section 3.3.3).

1860 Each implementation is free to decide the "best" action to take depending on local policy, whether any
1861 documents have been added, whether the implementation spools jobs or not, and/or any other piece of
1862 information available to it. If the choice is to abort the Job object, it is possible that the Job object may
1863 already have been processed to the point that some media sheet pages have been printed.

1864 3.3.1.1 Send-Document Request

1865 The following attribute sets are part of the Send-Document Request:

1866 Group 1: Operation Attributes

1867 Natural Language and Character Set:

1868 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1869 3.1.4.1.

1871 Target:

1872 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation
1873 attribute(s) which define the target for this operation as described in section 3.1.5.

1874

1875 Requesting User Name:

1876 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
1877 described in section 8.3.

1878

1879 "document-name" (name(MAX)):

1880 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1881 attribute. It contains the client supplied document name. The document name MAY be different
1882 than the Job name. It might be helpful, but NEED NOT be unique across multiple documents in
1883 the same Job. Typically, the client software automatically supplies the document name on behalf
1884 of the end user by using a file name or an application generated name. See the description of the
1885 "document-name" operation attribute in the Print-Job Request (section 3.2.1.1) for more
1886 information about this attribute.

1887

1888 "document-format" (mimeMediaType) :

1889 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1890 attribute. The value of this attribute identifies the format of the supplied document data. If the
1891 client does not supply this attribute, the Printer object assumes that the document data is in the
1892 format defined by the Printer object's "document-format-default" attribute. If the client supplies
1893 this attribute, but the value is not supported by the Printer object, i.e., the value is not one of the
1894 values of the Printer object's "document-format-supported" attribute, the Printer object MUST
1895 copy the attribute and its value to the Unsupported Attributes response group, reject the request
1896 and return the 'client-error-document-format-not-supported' status code. Issue 11

1897

1898 "document-natural-language" (naturalLanguage):

1899 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports
1900 this attribute. This attribute specifies the natural language of the document for those document-
1901 formats that require a specification of the natural language in order to image the document
1902 unambiguously. There are no particular values required for the Printer object to support.

1903

1904 "compression" (type3 keyword)

1905 The client OPTIONALLY supplies this attribute. The Printer object ~~OPTIONALLY~~**MUST**
1906 supports this attribute and the "compression-supported" attribute (see section 4.4.29). The client
1907 supplied "compression" operation attribute identifies the compression algorithm used on the
1908 document data. If the client omits this attribute, the Printer object MUST assume that the data is
1909 not compressed. If the client supplies the attribute and the Printer object supports the attribute
1910 value, the Printer object MUST use the corresponding decompression algorithm on the document
1911 data. If the client supplies this attribute, but the value is not supported by the Printer object, i.e.,
1912 the value is not one of the values of the Printer object's "compression-supported" attribute, the
1913 Printer object MUST copy the attribute and its value to the Unsupported Attributes response
1914 group, reject the request, and return the 'client-error-attributes-or-values-not-supported' status
1915 code. Issue 28

1916

1917 "last-document" (boolean):

1918 The client MUST supply this attribute. The Printer object MUST support this attribute. It is a
1919 boolean flag that is set to 'true' if this is the last document for the Job, 'false' otherwise.

1920

1921 Group 2: Document Content

1922 The client **MUST** supply the document data if the "last-document" flag is set to 'false'. However,
1923 since a client might not know that the previous document sent with a Send-Document (or Send-
1924 URI) operation was the last document (i.e., the "last-document" attribute was set to 'false'), it is
1925 legal to send a Send-Document request with no document data where the "last-document" flag is
1926 set to 'true'. Such a request **MUST NOT** increment the value of the Job object's "number-of-
1927 documents" attribute, since no real document was added to the job.

1928 3.3.1.2 Send-Document Response

1929 The following sets of attributes are part of the Send-Document Response:

1930 Group 1: Operation Attributes

1931 Status Message:

1932 In addition to the **REQUIRED** status code returned in every response, the response
1933 **OPTIONALLY** includes a "status-message" (text) operation attribute as described in sections 14
1934 and 3.1.6.

1935

1936 Natural Language and Character Set:

1937 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1938 3.1.4.2.

1939

1940 Group 2: Unsupported Attributes

1941 This is a set of Operation attributes supplied by the client (in the request) that are not supported
1942 by the Printer object or that conflict with one another (see sections 3.2.1.2 and the Implementer's
1943 Guide [IPP-IIG]). If the Printer object is not returning any Unsupported Attributes in the
1944 response, the Printer object **SHOULD** omit Group 2 rather than sending an empty group.
1945 However, a client **MUST** be able to accept an empty group.

1946

1947 Group 3: Job Object Attributes

1948 This is the same set of attributes as described in the Print-Job response (see section 3.2.1.2).

1949

1950 3.3.2 Send-URI Operation

1951 This **OPTIONAL** operation is identical to the Send-Document operation (see section 3.3.1) except that a
1952 client **MUST** supply a URI reference ("document-uri" operation attribute) rather than the document data
1953 itself. If a Printer object supports this operation, clients can use both Send-URI or Send-Document
1954 operations to add new documents to an existing multi-document Job object. However, if a client needs
1955 to indicate that the previous Send-URI or Send-Document was the last document, the client **MUST** use
1956 the Send-Document operation with no document data and the "last-document" flag set to 'true' (rather
1957 than using a Send-URI operation with no "document-uri" operation attribute).

1958 If a Printer object supports this operation, it MUST also support the Print-URI operation (see section
1959 3.2.2).

1960 The Printer object MUST validate the syntax and URI scheme of the supplied URI before returning a
1961 response, just as in the Print-URI operation.

1962 3.3.3 Cancel-Job Operation

1963 This REQUIRED operation allows a client to cancel a Print Job from the time the job is created up to the
1964 time it is completed, canceled, or aborted. Since a Job might already be printing by the time a Cancel-
1965 Job is received, some media sheet pages might be printed before the job is actually terminated.

1966 3.3.3.1 Cancel-Job Request

1967 The following groups of attributes are part of the Cancel-Job Request:

1968 Group 1: Operation Attributes

1969 Natural Language and Character Set:

1970 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1971 3.1.4.1.

1972

1973 Target:

1974 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri)
1975 operation attribute(s) which define the target for this operation as described in section 3.1.5.

1976

1977 Requesting User Name:

1978 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
1979 described in section 8.3.

1980

1981 "message" (text(127)):

1982 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports
1983 this attribute. It is a message to the operator. This "message" attribute is not the same as the "job-
1984 message-from-operator" attribute. That attribute is used to report a message from the operator to
1985 the end user that queries that attribute. This "message" operation attribute is used to send a
1986 message from the client to the operator along with the operation request. It is an implementation
1987 decision of how or where to display this message to the operator (if at all).

1988

1989 3.3.3.2 Cancel-Job Response

1990 The following sets of attributes are part of the Cancel-Job Response:

1991 Group 1: Operation Attributes

1992 Status Message:

1993 In addition to the REQUIRED status code returned in every response, the response
1994 OPTIONALLY includes a "status-message" (text) operation attribute as described in sections 14
1995 and 3.1.6.

1996
1997 If the job is already in the 'completed', 'aborted', or 'canceled' state, or the 'process-to-stop-point'
1998 value is set in the Job's "job-state-reasons" attribute, the Printer object MUST reject the request
1999 and return the 'client-error-not-possible' error status code.

2000
2001 Natural Language and Character Set:

2002 The "attributes-charset" and "attributes-natural-language" attributes as described in section
2003 3.1.4.2.

2004

2005 Group 2: Unsupported Attributes

2006 This is a set of Operation attributes supplied by the client (in the request) that are not supported
2007 by the Printer object or that conflict with one another (see section 3.2.1.2 and the Implementer's
2008 Guide [IPP-IIG]). If the Printer object is not returning any Unsupported Attributes in the
2009 response, the Printer object SHOULD omit Group 2 rather than sending an empty group.
2010 However, a client MUST be able to accept an empty group.

2011

2012 Once a successful response has been sent, the implementation guarantees that the Job will eventually end
2013 up in the 'canceled' state. Between the time of the Cancel-Job operation is accepted and when the job
2014 enters the 'canceled' job-state (see section 4.3.7), the "job-state-reasons" attribute SHOULD contain the '
2015 processing-to-stop-point' value which indicates to later queries that although the Job might still be
2016 'processing', it will eventually end up in the 'canceled' state, not the 'completed' state.

2017 3.3.4 Get-Job-Attributes Operation

2018 This REQUIRED operation allows a client to request the values of attributes of a Job object and it is
2019 almost identical to the Get-Printer-Attributes operation (see section 3.2.5). The only differences are that
2020 the operation is directed at a Job object rather than a Printer object, there is no "document-format"
2021 operation attribute used when querying a Job object, and the returned attribute group is a set of Job
2022 object attributes rather than a set of Printer object attributes.

2023 For Jobs, the possible names of attribute groups are:

- 2024 - 'job-template': all of the Job Template attributes that apply to a Job object (the first column of the
2025 table in Section 4.2).
- 2026 - 'job-description': all of the Job Description attributes specified in Section 4.3.
- 2027 - 'all': the special group 'all' that includes all supported attributes.

2028

2029 Since a client MAY request specific attributes or named groups, there is a potential that there is some
2030 overlap. For example, if a client requests, 'job-name' and 'job-description', the client is actually

2031 requesting the "job-name" attribute once by naming it explicitly, and once by inclusion in the 'job-
2032 description' group. In such cases, the Printer object NEED NOT return the attribute only once in the
2033 response even if it is requested multiple times. The client SHOULD NOT request the same attribute in
2034 multiple ways.

2035 It is NOT REQUIRED that a Job object support all attributes belonging to a group (since some attributes
2036 are OPTIONAL). However it is REQUIRED that each Job object support all group names.

2037 3.3.4.1 Get-Job-Attributes Request

2038 The following groups of attributes are part of the Get-Job-Attributes Request when the request is
2039 directed at a Job object:

2040 Group 1: Operation Attributes

2041 Natural Language and Character Set:

2042 The "attributes-charset" and "attributes-natural-language" attributes as described in section
2043 3.1.4.1.

2044 Target:

2045 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri)
2046 operation attribute(s) which define the target for this operation as described in section 3.1.5.

2047 Requesting User Name:

2048 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
2049 described in section 8.3.

2050 "requested-attributes" (1setOf keyword) :

2051 The client OPTIONALLY supplies this attribute. The IPP object MUST support this attribute.
2052 It is a set of attribute names and/or attribute group names in whose values the requester is
2053 interested. If the client omits this attribute, the IPP object MUST respond as if this attribute had
2054 been supplied with a value of 'all'.

2055 3.3.4.2 Get-Job-Attributes Response

2056 The Printer object returns the following sets of attributes as part of the Get-Job-Attributes Response:

2057 Group 1: Operation Attributes

2058 Status Message:

2059 In addition to the REQUIRED status code returned in every response, the response
2060 OPTIONALLY includes a "status-message" (text) operation attribute as described in sections 14
2061 and 3.1.6.

2067 Natural Language and Character Set:

2068 The "attributes-charset" and "attributes-natural-language" attributes as described in section
 2069 3.1.4.2. The "attributes-natural-language" MAY be the natural language of the Job object, rather
 2070 than the one requested.

2071

2072 Group 2: Unsupported Attributes

2073 This is a set of Operation attributes supplied by the client (in the request) that are not supported
 2074 by the Printer object or that conflict with one another (see sections 3.2.1.2 and the Implementer's
 2075 Guide [IPP-IIG]). The response NEED NOT contain the "requested-attributes" operation
 2076 attribute with any supplied values (attribute keywords) that were requested by the client but are
 2077 not supported by the IPP object. **If the Printer does include unsupported attributes referenced in**
 2078 **"requested-attributes" and such attributes include group names, such as "all", the unsupported**
 2079 **attributes MUST NOT include attributes described in the standard but not supported by the**
 2080 **implementation. Issue 23** -If the Printer object is not returning any Unsupported Attributes in
 2081 the response, the Printer object SHOULD omit Group 2 rather than sending an empty group.
 2082 However, a client MUST be able to accept an empty group.

2083

2084 Group 3: Job Object Attributes

2085 This is the set of requested attributes and their current values. The IPP object ignores (does not
 2086 respond with) any requested attribute or value which is not supported or which is restricted by the
 2087 security policy in force, including whether the requesting user is the user that submitted the job
 2088 (job originating user) or not (see section 8). However, the IPP object MUST respond with the
 2089 'unknown' value for any supported attribute (including all REQUIRED attributes) for which the
 2090 IPP object does not know the value, unless it would violate the security policy. See the
 2091 description of the "out-of-band" values in the beginning of Section 4.1.

2092 3.3.5 Hold-Job Operation

2093 This OPTIONAL operation allows a client to hold a pending job in the queue so that it is not eligible for
 2094 scheduling. If the Hold-Job operation is supported, then the Release-Job operation MUST be supported,
 2095 and vice-versa. The OPTIONAL "job-hold-until" operation attribute allows a client to specify whether
 2096 to hold the job indefinitely or until a specified time period, if supported.

2097 The IPP object MUST accept or reject the request based on the job's current state and transition the job
 2098 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'

Current "job-state"	New "job-state"	IPP object's response status code and action:
'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'

2099 Note 1: If the OPTIONAL "job-state-reasons" attribute is supported and if the implementation supports
 2100 multiple reasons for a job to be in the 'pending-held' state, the IPP object MUST add the 'job-hold-until-
 2101 specified' value to the job's "job-state-reasons" attribute.

2102 Note 2: If the IPP object supports the "job-hold-until" operation attribute, but the specified time period
 2103 has already started (or is the 'no-hold' value) and there are no other reasons to hold the job, the IPP object
 2104 MUST make the job be a candidate for processing immediately (see Section 4.2.2) by putting the job in
 2105 the 'pending' state.

2106 Note: In order to keep the Hold-Job operation simple, such a request is rejected when the job is in the
 2107 'processing' or 'processing-stopped' states. If an operation is needed to hold jobs while in these states, it
 2108 will be added as an additional operation, rather than overloading the Hold-Job operation. Then it is clear
 2109 to clients by querying the Printer object's "operations-supported" (see Section 4.4.13) and the Job
 2110 object's "job-state" (see Section 4.3.7) attributes which operations are possible.

2111 *Access Rights:* The requesting user must either be the submitter of the job or an operator or administrator
 2112 of the Printer object (see Section 1). Otherwise, the IPP object MUST reject the operation and return:
 2113 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' as appropriate.

2114 3.3.5.1 Hold-Job Request

2115 The groups and operation attributes are the same as for a Cancel-Job request (see section 3.3.3.1), with
 2116 the addition of the following Group 1 Operation attribute:

2117 "job-hold-until" (type3 keyword | name(MAX)):

2118 The client OPTIONALLY supplies this Operation attribute. The IPP object MUST support this
 2119 operation attribute in a Hold-Job request, if it supports the "job-hold-until" Job template attribute
 2120 in create operations. See section 4.2.2. The IPP object SHOULD support the "job-hold-until"
 2121 Job Template attribute for use in job create operations with at least the 'indefinite' value, if it
 2122 supports the Hold-Job operation. Otherwise, a client cannot create a job and hold it immediately
 2123 (without picking some supported time period in the future).

2124 If supplied and supported as specified in the Printer's "job-hold-until-supported" attribute, the IPP
 2125 object copies the supplied operation attribute to the Job object, replacing the job's previous "job-
 2126 hold-until" attribute, if present, and makes the job a candidate for scheduling during the supplied
 2127 named time period.

2128 If supplied, but either the "job-hold-until" Operation attribute itself or the value supplied is not
 2129 supported, the IPP object accepts the request, returns the unsupported attribute or value in the
 2130 Unsupported Attributes Group according to section 3.2.1.2, returns the 'successful-ok-ignored-or-

2131 substituted-attributes, and holds the job indefinitely until a client performs a subsequent Release-
2132 Job operation.

2133 If the client (1) supplies a value that specifies a time period that has already started or the 'no-
2134 hold' value (meaning don't hold the job) and (2) the IPP object supports the "job-hold-until"
2135 operation attribute and there are no other reasons to hold the job, the IPP object MUST accept the
2136 operation and make the job be a candidate for processing immediately (see Section 4.2.2).

2137 If the client does not supply a "job-hold-until" Operation attribute in the request, the IPP object
2138 MUST populate the job object with a "job-hold-until" attribute with the 'indefinite' value (if IPP
2139 object supports the "job-hold-until" attribute) and hold the job indefinitely, until a client performs
2140 a Release-Job operation.

2141 3.3.5.2 Hold-Job Response

2142 The groups and attributes are the same as for a Cancel-Job response (see section 3.3.3.2).

2143 3.3.6 Release-Job Operation

2144 This OPTIONAL operation allows a client to release a previously held job so that it is again eligible for
2145 scheduling. If the Hold-Job operation is supported, then the Release-Job operation MUST be supported,
2146 and vice-versa.

2147 This operation removes the "job-hold-until" job attribute, if present, from the job object that had been
2148 supplied in the create or most recent Hold-Job or Restart-Job operation and remove its effect on the job.
2149 If the OPTIONAL "job-state-reasons" attribute is supported, the IPP object MUST remove the 'job-hold-
2150 until-specified' value from the job's "job-state-reasons" attribute, if present. See section 4.3.8.

2151 The IPP object MUST accept or reject the request based on the job's current state and transition the job
2152 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'

2153 Note 1: If there are other reasons to keep the job in the 'pending-held' state, such as 'resources-are-not-
2154 ready', the job remains in the 'pending-held' state. Thus the 'pending-held' state is not just for jobs that
2155 have the 'job-hold-until' applied to them, but are for any reason to keep the job from being a candidate

2156 for scheduling and processing, such as 'resources-are-not-ready'. See the "job-hold-until" attribute
2157 (section 4.2.2).

2158 *Access Rights:* The requesting user must either be the submitter of the job or an operator or administrator
2159 of the Printer object. Otherwise, the IPP object MUST reject the operation and return: 'client-error-
2160 forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' as appropriate.

2161 The Release-Job Request and Release-Job Response have the same attribute groups and attributes as the
2162 Cancel-Job operation (see section 3.3.3.1 and 3.3.3.2).

2163 3.3.7 Restart-Job Operation

2164 This OPTIONAL operation allows a client to restart a job that is retained in the queue after processing
2165 has completed (see section 4.3.7.1).

2166 The job is moved to the 'pending' job state and restarts at the beginning on the same IPP Printer object
2167 with the same attribute values. The Job Description attributes that accumulate job progress, such as
2168 "job-impressions-completed", "job-media-sheets-completed", and "job-k-octets-processed", MUST be
2169 reset to 0 so that they give an accurate record of the job from its restart point. The job object MUST
2170 continue to use the same "job-uri" and "job-id" attribute values.

2171 Note: If in the future an operation is needed that does not reset the job progress attributes, then a new
2172 operation will be defined which makes a copy of the job, assigns a new "job-uri" and "job-id" to the copy
2173 and resets the job progress attributes in the new copy only.

2174 The IPP object MUST accept or reject the request based on the job's current state, transition the job to
2175 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'		
'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

2176

2177 Note 1: If the Job Retention Period has expired for the job in this state, then the IPP object rejects the
2178 operation. See section 4.3.7.1.

2179 Note: In order to prevent a user from inadvertently restarting a job in the middle, the Restart-Job request
2180 is rejected when the job is in the 'processing' or 'processing-stopped' states. If in the future an operation
2181 is needed to hold or restart jobs while in these states, it will be added as an additional operation, rather
2182 than overloading the Restart-Job operation, so that it is clear that the user intended that the current job
2183 not be completed.

2184 *Access Rights:* The requesting user must either be the submitter of the job or an operator or administrator
2185 of the Printer object. Otherwise, the IPP object MUST reject the operation and return: 'client-error-
2186 forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' as appropriate.

2187 3.3.7.1 Restart-Job Request

2188 The groups and attributes are the same as for a Cancel-Job request (see section 3.3.3.1), with the addition
2189 of the following Group 1 Operation attribute:

2190 "job-hold-until" (type3 keyword | name(MAX)):

2191 The client OPTIONALLY supplies this attribute. The IPP object MUST support this Operation
2192 attribute in a Restart-Job request, if it supports the "job-hold-until" Job Template attribute in
2193 create operations. See section 4.2.2. Otherwise, the IPP object NEED NOT support the "job-
2194 hold-until" Operation attribute in a Restart-Job request.

2195 If supplied and supported as specified in the Printer's "job-hold-until-supported" attribute, the IPP
2196 object copies the supplied Operation attribute to the Job object, replacing the job's previous "job-
2197 hold-until" attribute, if present, and makes the job a candidate for scheduling during the supplied
2198 named time period. See section 4.2.2.

2199 If supplied, but the value is not supported, the IPP object accepts the request, returns the
2200 unsupported attribute or value in the Unsupported Attributes Group according to section 3.2.1.2,
2201 returns the 'successful-ok-ignored-or-substituted-attributes' status code, and holds the job
2202 indefinitely until a client performs a subsequent Release-Job operation.

2203 If supplied, but the "job-hold-until" Operation attribute itself is not supported, the IPP object
2204 accepts the request, returns the unsupported attribute with the out-of-band 'unsupported' value in
2205 the Unsupported Attributes Group according to section 3.2.1.2, returns the 'successful-ok-
2206 ignored-or-substituted-attributes' status code, and restarts the job, i.e., ignores the "job-hold-
2207 until" attribute.

2208 If the client (1) supplies a value that specifies a time period that has already started or the 'no-
2209 hold' value (meaning don't hold the job) and (2) the IPP object supports the "job-hold-until"
2210 operation attribute and there are no other reasons to hold the job, the IPP object makes the job a
2211 candidate for processing immediately (see Section 4.2.2).

2212 If the client does not supply a "job-hold-until" operation attribute in the request, the IPP object
2213 removes the "job-hold-until" attribute, if present, from the job. If there are no other reasons to
2214 hold the job, the Restart-Job operation makes the job a candidate for processing immediately (see
2215 Section 4.2.2).

2216 3.3.7.2 Restart-Job Response

2217 The groups and attributes are the same as for a Cancel-Job response (see section 3.3.3.2).

2218 Note: In the future an OPTIONAL Modify-Job operation may be specified that allows the client to
2219 modify other attributes before releasing the restarted job.

2220 4. Object Attributes

2221 This section describes the attributes with their corresponding attribute syntaxes and values that are part
2222 of the IPP model. The sections below show the objects and their associated attributes which are
2223 included within the scope of this protocol. Many of these attributes are derived from other relevant
2224 specifications:

2225 - Document Printing Application (DPA) [ISO10175]

2226 - RFC 1759 Printer MIB [RFC1759]

2227

2228 Each attribute is uniquely identified in this document using a "keyword" (see section 13.2.1) which is the
2229 name of the attribute. The keyword is included in the section header describing that attribute.

2230 Note: Not only are keywords used to identify attributes, but one of the attribute syntaxes described
2231 below is "keyword" so that some attributes have keyword values. Therefore, these attributes are defined
2232 as having an attribute syntax that is a set of keywords.

2233 4.1 Attribute Syntaxes

2234 This section defines the basic attribute syntax types that all clients and IPP objects MUST be able to
2235 accept in responses and accept in requests, respectively. Each attribute description in sections 3 and
2236 3.3.5 includes the name of attribute syntax(es) in the heading (in parentheses). A conforming
2237 implementation of an attribute MUST include the semantics of the attribute syntax(es) so identified.
2238 Section 6.3 describes how the protocol can be extended with new attribute syntaxes.

2239 The attribute syntaxes are specified in the following sub-sections, where the sub-section heading is the
2240 keyword name of the attribute syntax inside the single quotes. In operation requests and responses each
2241 attribute value MUST be represented as one of the attribute syntaxes specified in the sub-section heading
2242 for the attribute. In addition, the value of an attribute in a response (but not in a request) MAY be one of
2243 the "out-of-band" values whose special encoding rules are defined in the "Encoding and Transport"
2244 specification [IPP-PRO]. Standard "out-of-band" values are: **Issues 12 and 15**

2245 'unknown': The attribute is supported by the IPP object, but the value is unknown to the IPP object
2246 for some reason.

2247 'unsupported': The attribute is unsupported by the IPP object. This value MUST be returned only as
2248 the value of an attribute in the Unsupported Attributes Group.

2249 'no-value': The attribute is supported by the Printer object, but the administrator has not yet
2250 configured a value.

2251

2252 ~~The "Encoding and Transport" specification [IPP-PRO] defines mechanisms for passing "out-of-band"~~
2253 ~~values.~~ All attributes in a request MUST have one or more values as defined in Sections 4.2 to 4.4.
2254 Thus clients MUST NOT supply attributes with "out-of-band" values. All attributes in a response
2255 MUST have one or more values as defined in Sections 4.2 to 4.4 or a single "out-of-band" value.

2256 Most attributes are defined to have a single attribute syntax. However, a few attributes (e.g., "job-sheet",
2257 "media", "job-hold-until") are defined to have several attribute syntaxes, depending on the value. These
2258 multiple attribute syntaxes are separated by the "|" character in the sub-section heading to indicate the
2259 choice. Since each value MUST be tagged as to its attribute syntax in the protocol, a single-valued
2260 attribute instance may have any one of its attribute syntaxes and a multi-valued attribute instance may
2261 have a mixture of its defined attribute syntaxes.

2262 4.1.1 'text'

2263 A text attribute is an attribute whose value is a sequence of zero or more characters encoded in a
2264 maximum of 1023 (MAX) octets. MAX is the maximum length for each value of any text attribute.
2265 However, if an attribute will always contain values whose maximum length is much less than MAX, the
2266 definition of that attribute will include a qualifier that defines the maximum length for values of that
2267 attribute. For example: the "printer-location" attribute is specified as "printer-location (text(127))". In
2268 this case, text values for "printer-location" MUST NOT exceed 127 octets; if supplied with a longer text
2269 string via some external interface (other than the protocol), implementations are free to truncate to this
2270 shorter length limitation.

2271 In this specification, all text attributes are defined using the 'text' syntax. However, 'text' is used only for
2272 brevity; the formal interpretation of 'text' is: 'textWithoutLanguage | textWithLanguage'. That is, for any
2273 attribute defined in this specification using the 'text' attribute syntax, all IPP objects and clients MUST
2274 support both the 'textWithoutLanguage' and 'textWithLanguage' attribute syntaxes. However, in actual
2275 usage and protocol execution, objects and clients accept and return only one of the two syntax per
2276 attribute. The syntax 'text' never appears "on-the-wire".

2277 Both 'textWithoutLanguage' and 'textWithLanguage' are needed to support the real world needs of
2278 interoperability between sites and systems that use different natural languages as the basis for human
2279 communication. Generally, one natural language applies to all text attributes in a given request or
2280 response. The language is indicated by the "attributes-natural-language" operation attribute defined in
2281 section 3.1.4 or "attributes-natural-language" job attribute defined in section 4.3.24, and there is no need
2282 to identify the natural language for each text string on a value-by-value basis. In these cases, the
2283 attribute syntax 'textWithoutLanguage' is used for text attributes. In other cases, the client needs to
2284 supply or the Printer object needs to return a text value in a natural language that is different from the
2285 rest of the text values in the request or response. In these cases, the client or Printer object uses the
2286 attribute syntax 'textWithLanguage' for text attributes (this is the Natural Language Override mechanism
2287 described in section 3.1.4).

2288 The 'textWithoutLanguage' and 'textWithLanguage' attribute syntaxes are described in more detail in the
2289 following sections.

2290 4.1.1.1 'textWithoutLanguage'

2291 The 'textWithoutLanguage' syntax indicates a value that is sequence of zero or more characters. Text
2292 strings are encoded using the rules of some charset. The Printer object MUST support the UTF-8
2293 charset [RFC2279] and MAY support additional charsets to represent 'text' values, provided that the
2294 charsets are registered with IANA [IANA-CS]. See Section 4.1.7 for the specification of the 'charset'
2295 attribute syntax, including restricted semantics and examples of charsets.

2296 4.1.1.2 'textWithLanguage'

2297 The 'textWithLanguage' attribute syntax is a compound attribute syntax consisting of two parts: a
2298 'textWithoutLanguage' part plus an additional 'naturalLanguage' (see section 4.1.8) part that overrides the
2299 natural language in force. The 'naturalLanguage' part explicitly identifies the natural language that
2300 applies to the text part of that value and that value alone. For any give text attribute, the
2301 'textWithoutLanguage' part is limited to the maximum length defined for that attribute, but the
2302 'naturalLanguage' part is always limited to 63 octets. Using the 'textWithLanguage' attribute syntax rather
2303 than the normal 'textWithoutLanguage' syntax is the so-called Natural Language Override mechanism
2304 and MUST be supported by all IPP objects and clients.

2305 If the attribute is multi-valued (1setOf text), then the 'textWithLanguage' attribute syntax MUST be used
2306 to explicitly specify each attribute value whose natural language needs to be overridden. Other values in
2307 a multi-valued 'text' attribute in a request or a response revert to the natural language of the operation
2308 attribute.

2309 In a create request, the Printer object MUST accept and store with the Job object any natural language in
2310 the "attributes-natural-language" operation attribute, whether the Printer object supports that natural
2311 language or not. Furthermore, the Printer object MUST accept and store any 'textWithLanguage'
2312 attribute value, whether the Printer object supports that natural language or not. These requirements are
2313 independent of the value of the "ipp-attribute-fidelity" operation attribute that the client MAY supply.

2314 Example: If the client supplies the "attributes-natural-language" operation attribute with the value: 'en'
2315 indicating English, but the value of the "job-name" attribute is in French, the client MUST use the
2316 'textWithLanguage' attribute syntax with the following two values:

2317 'fr': Natural Language Override indicating French

2318 'Rapport Mensuel': the job name in French

2319

2320 See the "Encoding and Transport" document [IPP-PRO] for a detailed example of the
2321 'textWithLanguage' attribute syntax.

2322 4.1.2 'name'

2323 This syntax type is used for user-friendly strings, such as a Printer name, that, for humans, are more
2324 meaningful than identifiers. Names are never translated from one natural language to another. The
2325 'name' attribute syntax is essentially the same as 'text', including the REQUIRED support of UTF-8

2326 except that the sequence of characters is limited so that its encoded form MUST NOT exceed 255
2327 (MAX) octets.

2328 Also like 'text', 'name' is really an abbreviated notation for either 'nameWithoutLanguage' or
2329 'nameWithLanguage'. That is, all IPP objects and clients MUST support both the
2330 'nameWithoutLanguage' and 'nameWithLanguage' attribute syntaxes. However, in actual usage and
2331 protocol execution, objects and clients accept and return only one of the two syntax per attribute. The
2332 syntax 'name' never appears "on-the-wire".

2333 Note: Only the 'text' and 'name' attribute syntaxes permit the Natural Language Override mechanism.

2334 Some attributes are defined as 'type3 keyword | name'. These attributes support values that are either
2335 type3 keywords or names. This dual-syntax mechanism enables a site administrator to extend these
2336 attributes to legally include values that are locally defined by the site administrator. Such names are not
2337 registered with IANA.

2338 4.1.2.1 'nameWithoutLanguage'

2339 The 'nameWithoutLanguage' syntax indicates a value that is sequence of zero or more characters so that
2340 its encoded form does not exceed MAX octets.

2341 4.1.2.2 'nameWithLanguage'

2342 The 'nameWithLanguage' attribute syntax is a compound attribute syntax consisting of two parts: a
2343 'nameWithoutLanguage' part plus an additional 'naturalLanguage' (see section 4.1.8) part that overrides
2344 the natural language in force. The 'naturalLanguage' part explicitly identifies the natural language that
2345 applies to that name value and that name value alone.

2346 The 'nameWithLanguage' attribute syntax behaves the same as the 'textWithLanguage' syntax. If a name
2347 is in a language that is different than the rest of the object or operation, then this 'nameWithLanguage'
2348 syntax is used rather than the generic 'nameWithoutLanguage' syntax.

2349 Example: If the client supplies the "attributes-natural-language" operation attribute with the value: 'en'
2350 indicating English, but the "printer-name" attribute is in German, the client MUST use the
2351 'nameWithLanguage' attribute syntax as follows:

2352 'de': Natural Language Override indicating German

2353 'Farbdrucker': the Printer name in German

2354

2355 4.1.2.3 Matching 'name' attribute values

2356 For purposes of matching two 'name' attribute values for equality, such as in job validation (where a
2357 client-supplied value for attribute "xxx" is checked to see if the value is among the values of the Printer
2358 object's corresponding "xxx-supported" attribute), the following match rules apply:

2359 1. 'keyword' values never match 'name' values.

2360 2. 'name' (nameWithoutLanguage and nameWithLanguage) values match if (1) the name parts
2361 match and (2) the Associated Natural-Language parts (see section 3.1.4.1) match. The matching
2362 rules are:

2363 a. the name parts match if the two names are identical character by character, except it is
2364 RECOMMENDED that case be ignored. For example: 'Ajax-letter-head-white' MUST
2365 match 'Ajax-letter-head-white' and SHOULD match 'ajax-letter-head-white' and 'AJAX-
2366 LETTER-HEAD-WHITE'.

2367 b. the Associated Natural-Language parts match if the shorter of the two meets the
2368 syntactic requirements of RFC 1766 [RFC1766] and matches byte for byte with the
2369 longer. For example, 'en' matches 'en', 'en-us' and 'en-gb', but matches neither 'fr' nor 'e'.

2370 4.1.3 'keyword'

2371 The 'keyword' attribute syntax is a sequence of characters, length: 1 to 255, containing only the US-
2372 ASCII [ASCII] encoded values for lowercase letters ("a" - "z"), digits ("0" - "9"), hyphen ("-"), dot ("."),
2373 and underscore ("_"). The first character MUST be a lowercase letter. Furthermore, keywords MUST
2374 be in U.S. English.

2375 This syntax type is used for enumerating semantic identifiers of entities in the abstract protocol, i.e.,
2376 entities identified in this document. Keywords are used as attribute names or values of attributes.
2377 Unlike 'text' and 'name' attribute values, 'keyword' values MUST NOT use the Natural Language
2378 Override mechanism, since they MUST always be US-ASCII and U.S. English.

2379 Keywords are for use in the protocol. A user interface will likely provide a mapping between protocol
2380 keywords and displayable user-friendly words and phrases which are localized to the natural language of
2381 the user. While the keywords specified in this document MAY be displayed to users whose natural
2382 language is U.S. English, they MAY be mapped to other U.S. English words for U.S. English users,
2383 since the user interface is outside the scope of this document.

2384 In the definition for each attribute of this syntax type, the full set of defined keyword values for that
2385 attribute are listed.

2386 When a keyword is used to represent an attribute (its name), it MUST be unique within the full scope of
2387 all IPP objects and attributes. When a keyword is used to represent a value of an attribute, it MUST be
2388 unique just within the scope of that attribute. That is, the same keyword MUST NOT be used for two
2389 different values within the same attribute to mean two different semantic ideas. However, the same
2390 keyword MAY be used across two or more attributes, representing different semantic ideas for each
2391 attribute. Section 6.1 describes how the protocol can be extended with new keyword values. Examples
2392 of attribute name keywords:

2393 "job-name"
2394 "attributes-charset"
2395

2396 Note: This document uses "type1", "type2", and "type3" prefixes to the "keyword" basic syntax to
2397 indicate different levels of review for extensions (see section 6.1).

2398 4.1.4 'enum'

2399 The 'enum' attribute syntax is an enumerated integer value that is in the range from 1 to $2^{31} - 1$
2400 (MAX). Each value has an associated 'keyword' name. In the definition for each attribute of this syntax
2401 type, the full set of possible values for that attribute are listed. This syntax type is used for attributes for
2402 which there are enum values assigned by other standards, such as SNMP MIBs. A number of attribute
2403 enum values in this specification are also used for corresponding attributes in other standards
2404 [RFC1759]. This syntax type is not used for attributes to which the administrator may assign values.
2405 Section 6.1 describes how the protocol can be extended with new enum values.

2406 Enum values are for use in the protocol. A user interface will provide a mapping between protocol enum
2407 values and displayable user-friendly words and phrases which are localized to the natural language of the
2408 user. While the enum symbols specified in this document MAY be displayed to users whose natural
2409 language is U.S. English, they MAY be mapped to other U.S. English words for U.S. English users,
2410 since the user interface is outside the scope of this document.

2411 Note: SNMP MIBs use '2' for 'unknown' which corresponds to the IPP "out-of-band" value 'unknown'.
2412 See the description of the "out-of-band" values at the beginning of Section 4.1. Therefore, attributes of
2413 type 'enum' start at '3'.

2414 Note: This document uses "type1", "type2", and "type3" prefixes to the "enum" basic syntax to indicate
2415 different levels of review for extensions (see section 6.1).

2416 4.1.5 'uri'

2417 The 'uri' attribute syntax is any valid Uniform Resource Identifier or URI [RFC2396]. Most often, URIs
2418 are simply Uniform Resource Locators or URLs. The maximum length of URIs used as values of IPP
2419 attributes is 1023 octets. Although most other IPP attribute syntax types allow for only lower-cased
2420 values, this attribute syntax type conforms to the case-sensitive and case-insensitive rules specified in
2421 [RFC2396]. See also [IPP-IIG] for a discussion of case in URIs.

2422 4.1.6 'uriScheme'

2423 The 'uriScheme' attribute syntax is a sequence of characters representing a URI scheme according to
2424 RFC 2396 [RFC2396]. Though RFC 2396 requires that the values be case-insensitive, IPP requires all
2425 lower case values in IPP attributes to simplify comparing by IPP clients and Printer objects. Standard
2426 values for this syntax type are the following keywords:

2427 'http': for HTTP schemed URIs (e.g., "http:...")
2428 'https': for use with HTTPS schemed URIs (e.g., "https:...") (not on IETF standards track)
2429 'ftp': for FTP schemed URIs (e.g., "ftp:...")
2430 'mailto': for SMTP schemed URIs (e.g., "mailto:...")
2431 'file': for file schemed URIs (e.g., "file:...")

2432

2433 A Printer object MAY support any URI 'scheme' that has been registered with IANA [IANA-MT]. The
2434 maximum length of URI 'scheme' values used to represent IPP attribute values is 63 octets.

2435 4.1.7 'charset'

2436 The 'charset' attribute syntax is a standard identifier for a charset. A charset is a coded character set and
2437 encoding scheme. Charsets are used for labeling certain document contents and 'text' and 'name'
2438 attribute values. The syntax and semantics of this attribute syntax are specified in RFC 2046 [RFC2046]
2439 and contained in the IANA character-set Registry [IANA-CS] according to the IANA procedures
2440 [RFC2278]. Though RFC 2046 requires that the values be case-insensitive US-ASCII, IPP requires all
2441 lower case values in IPP attributes to simplify comparing by IPP clients and Printer objects. When a
2442 character-set in the IANA registry has more than one name (alias), the name labeled as "(preferred
2443 MIME name)", if present, MUST be used.

2444 The maximum length of 'charset' values used to represent IPP attribute values is 63 octets.

2445 Some examples are:

2446 'utf-8': ISO 10646 Universal Multiple-Octet Coded Character Set (UCS) represented as the UTF-8
2447 [RFC2279] transfer encoding scheme in which US-ASCII is a subset charset.

2448 'us-ascii': 7-bit American Standard Code for Information Interchange (ASCII), ANSI X3.4-1986
2449 [ASCII]. That standard defines US-ASCII, but RFC 2045 [RFC2045] eliminates most of the
2450 control characters from conformant usage in MIME and IPP.

2451 'iso-8859-1': 8-bit One-Byte Coded Character Set, Latin Alphabet Nr 1 [ISO8859-1]. That standard
2452 defines a coded character set that is used by Latin languages in the Western Hemisphere and
2453 Western Europe. US-ASCII is a subset charset.

2454 'iso-10646-ucs-2': ISO 10646 Universal Multiple-Octet Coded Character Set (UCS) represented as
2455 two octets (UCS-2), with the high order octet of each pair coming first (so-called Big Endian
2456 integer).

2457

2458 Some attribute descriptions MAY place additional requirements on charset values that may be used, such
2459 as REQUIRED values that MUST be supported or additional restrictions, such as requiring that the
2460 charset have US-ASCII as a subset charset.

2461 4.1.8 'naturalLanguage'

2462 The 'naturalLanguage' attribute syntax is a standard identifier for a natural language and optionally a
2463 country. The values for this syntax type are defined by RFC 1766 [RFC1766]. Though RFC 1766
2464 requires that the values be case-insensitive US-ASCII, IPP requires all lower case to simplify comparing
2465 by IPP clients and Printer objects. Examples include:

2466 'en': for English

2467 'en-us': for US English

2468 'fr': for French

2469 'de': for German

2470

2471 The maximum length of 'naturalLanguage' values used to represent IPP attribute values is 63 octets.

2472 4.1.9 'mimeType'

2473 The 'mimeType' attribute syntax is the Internet Media Type (sometimes called MIME type) as
2474 defined by RFC 2046 [RFC2046] and registered according to the procedures of RFC 2048 [RFC2048]
2475 for identifying a document format. The value MAY include a charset parameter, depending on the
2476 specification of the Media Type in the IANA Registry [IANA-MT]. Although most other IPP syntax
2477 types allow for only lower-cased values, this syntax type allows for mixed-case values which are case-
2478 insensitive.

2479 Examples are:

2480 'text/html': An HTML document

2481 'text/plain': A plain text document in US-ASCII (RFC 2046 indicates that in the absence of the
2482 charset parameter MUST mean US-ASCII rather than simply unspecified) [RFC2046].

2483 'text/plain; charset=US-ASCII': A plain text document in US-ASCII [52, 56].

2484 'text/plain; charset=ISO-8859-1': A plain text document in ISO 8859-1 (Latin 1) [ISO8859-1].

2485 'text/plain; charset=utf-8': A plain text document in ISO 10646 represented as UTF-8 [RFC2279]

2486 'application/postscript': A PostScript document [RFC2046]

2487 'application/vnd.hp-PCL': A PCL document [IANA-MT] (charset escape sequence embedded in the
2488 document data)

2489 'application/pdf': Portable Document Format - see IANA MIME Media Type registry

2490 'application/octet-stream': ~~(REQUIRED)~~ Auto-sense - see below

2491

2492

2493 One special type is 'application/octet-stream'. If the Printer object supports this value, the Printer object
2494 MUST be capable of auto-sensing the format of the document data. Depending on when the Printer
2495 performs auto-sensing, it may detect auto-sensing errors before sending an operation response or
2496 afterwards. If the printer detects an error during auto-sensing, it adds the 'unsupported-document-format'
2497 value to the "printer-state-reasons". If the printer determines an auto-sensing error before sending a
2498 response, the response contains the printer-state-reasons and a status which specifies the error. Issues 9
2499 and 10.

2500 If the Printer object's default value attribute "document-format-default" is set to 'application/octet-
2501 stream', the Printer object not only supports auto-sensing of the document format, but will depend on the
2502 result of applying its auto-sensing when the client does not supply the "document-format" attribute. If
2503 the client supplies a document format value, the Printer MUST rely on the supplied attribute, rather than
2504 trust its auto-sensing algorithm. To summarize:

2505 1. If the client does not supply a document format value, the Printer MUST rely on its default value
2506 setting (which may be 'application/octet-stream' indicating an auto-sensing mechanism).

2507 2. If the client supplies a value other than 'application/octet-stream', the client is supplying valid
2508 information about the format of the document data and the Printer object MUST trust the client
2509 supplied value more than the outcome of applying an automatic format detection mechanism.

2510 For example, the client may be requesting the printing of a PostScript file as a 'text/plain'
2511 document. The Printer object MUST print a text representation of the PostScript commands
2512 rather than interpret the stream of PostScript commands and print the result.

2513 3. If the client supplies a value of 'application/octet-stream', the client is indicating that the Printer
2514 object MUST use its auto-sensing mechanism on the client supplied document data whether
2515 auto-sensing is the Printer object's default or not.
2516

2517 Note: Since the auto-sensing algorithm is probabilistic, if the client requests both auto-sensing
2518 ("document-format" set to 'application/octet-stream') and true fidelity ("ipp-attribute-fidelity" set to
2519 'true'), the Printer object might not be able to guarantee exactly what the end user intended (the auto-
2520 sensing algorithm might mistake one document format for another), but it is able to guarantee that its
2521 auto-sensing mechanism be used.

2522 The maximum length of a 'mimeType' value to represent IPP attribute values is 255 octets.

2523 4.1.10 'octetString'

2524 The 'octetString' attribute syntax is a sequence of octets encoded in a maximum of 1023 octets which is
2525 indicated in sub-section headers using the notation: octetString(MAX). This syntax type is used for
2526 opaque data.

2527 4.1.11 'boolean'

2528 The 'boolean' attribute syntax has only two values: 'true' and 'false'.

2529 4.1.12 'integer'

2530 The 'integer' attribute syntax is an integer value that is in the range from -2^{31} (MIN) to $2^{31} - 1$
2531 (MAX). Each individual attribute may specify the range constraint explicitly in sub-section headers if
2532 the range is different from the full range of possible integer values. For example: job-priority
2533 (integer(1:100)) for the "job-priority" attribute. However, the enforcement of that additional constraint is
2534 up to the IPP objects, not the protocol.

2535 4.1.13 'rangeOfInteger'

2536 The 'rangeOfInteger' attribute syntax is an ordered pair of integers that defines an inclusive range of
2537 integer values. The first integer specifies the lower bound and the second specifies the upper bound. If a
2538 range constraint is specified in the header description for an attribute in this document whose attribute
2539 syntax is 'rangeOfInteger' (i.e., 'X:Y' indicating X as a minimum value and Y as a maximum value), then
2540 the constraint applies to both integers.

2541 4.1.14 'dateTime'

2542 The 'dateTime' attribute syntax is a standard, fixed length, 11 octet representation of the "DateAndTime"
2543 syntax as defined in RFC 1903 [RFC1903]. RFC 1903 also identifies an 8 octet representation of a
2544 "DateAndTime" value, but IPP objects MUST use the 11 octet representation. A user interface will
2545 provide a mapping between protocol dateTime values and displayable user-friendly words or
2546 presentation values and phrases which are localized to the natural language and date format of the user.

2547 4.1.15 'resolution'

2548 The 'resolution' attribute syntax specifies a two-dimensional resolution in the indicated units. It consists
2549 of 3 values: a cross feed direction resolution (positive integer value), a feed direction resolution (positive
2550 integer value), and a units value. The semantics of these three components are taken from the Printer
2551 MIB [RFC1759] suggested values. That is, the cross feed direction component resolution component is
2552 the same as the prtMarkerAddressabilityXFeedDir object in the Printer MIB, the feed direction
2553 component resolution component is the same as the prtMarkerAddressabilityFeedDir in the Printer MIB,
2554 and the units component is the same as the prtMarkerAddressabilityUnit object in the Printer MIB
2555 (namely, '3' indicates dots per inch and '4' indicates dots per centimeter). All three values MUST be
2556 present even if the first two values are the same. Example: '300', '600', '3' indicates a 300 dpi cross-feed
2557 direction resolution, a 600 dpi feed direction resolution, since a '3' indicates dots per inch (dpi).

2558 4.1.16 '1setOf X'

2559 The '1setOf X' attribute syntax is 1 or more values of attribute syntax type X. This syntax type is used
2560 for multi-valued attributes. The syntax type is called '1setOf' rather than just 'setOf' as a reminder that
2561 the set of values MUST NOT be empty (i.e., a set of size 0). Sets are normally unordered. However
2562 each attribute description of this type may specify that the values MUST be in a certain order for that
2563 attribute.

2564 4.2 Job Template Attributes

2565 Job Template attributes describe job processing behavior. Support for Job Template attributes by a
2566 Printer object is OPTIONAL (see section 13.2.3 for a description of support for OPTIONAL attributes).
2567 Also, clients OPTIONALLY supply Job Template attributes in create requests.

2568 Job Template attributes conform to the following rules. For each Job Template attribute called "xxx":

- 2569 1. If the Printer object supports "xxx" then it MUST support both a "xxx-default" attribute (unless
2570 there is a "No" in the table below) and a "xxx-supported" attribute. If the Printer object doesn't
2571 support "xxx", then it MUST support neither an "xxx-default" attribute nor an "xxx-supported"
2572 attribute, and it MUST treat an attribute "xxx" supplied by a client as unsupported. An attribute
2573 "xxx" may be supported for some document formats and not supported for other document
2574 formats. For example, it is expected that a Printer object would only support "orientation-
2575 requested" for some document formats (such as 'text/plain' or 'text/html') but not others (such as
2576 'application/postscript').

2577

2578

2579

2580

2581

2582

2583

2584

2585

2586

2587

2588

2589

2590

2591

2592

2593

2594

2595

2596

2597

2598

2599

2600

2601

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

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

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

Note: The "xxx" in each "xxx-supported" attribute name is singular, even though an "xxx-supported" attribute usually has more than one value, such as "job-sheet-supported", unless the "xxx" Job Template attribute is plural, such as "finishings" or "sides". In such cases the "xxx-supported" attribute names are: "finishings-supported" and "sides-supported".

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

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

The "finishings" attribute is an example of a Job Template attribute. It can take on a set of values such as 'staple', 'punch', and/or 'cover'. A client can query the Printer object for the "finishings-supported" attribute and the "finishings-default" attribute. The supported attribute contains a set of supported values. The default value attribute contains the finishing value(s) that will be used for a new Job if the client does not supply a "finishings" attribute in the create request and the document data does not contain any corresponding finishing instructions. If the client does supply the "finishings" attribute in the create request, the IPP object validates the value or values to make sure that they are a subset of the supported values identified in the Printer object's "finishings-supported" attribute. See section 3.2.1.2.

The table below summarizes the names and relationships for all Job Template attributes. The first column of the table (labeled "Job Attribute") shows the name and syntax for each Job Template attribute

2620 in the Job object. These are the attributes that can optionally be supplied by the client in a create request.
2621 The last two columns (labeled "Printer: Default Value Attribute" and "Printer: Supported Values
2622 Attribute") shows the name and syntax for each Job Template attribute in the Printer object (the default
2623 value attribute and the supported values attribute). A "No" in the table means the Printer MUST NOT
2624 support the attribute (that is, the attribute is simply not applicable). For brevity in the table, the 'text' and
2625 'name' entries do not show the maximum length for each attribute.

2626			
2627	Job Attribute	Printer: Default Value Attribute	Printer: Supported Values Attribute
2628			
2629			
2630	job-priority (integer 1:100)	job-priority-default (integer 1:100)	job-priority-supported (integer 1:100)
2631			
2632			
2633	job-hold-until (type3 keyword name)	job-hold-until- default (type3 keyword name)	job-hold-until- supported (1setOf type3 keyword name)
2634			
2635			
2636			
2637			
2638	job-sheets (type3 keyword name)	job-sheets-default (type3 keyword name)	job-sheets-supported (1setOf type3 keyword name)
2639			
2640			
2641			
2642	multiple-document- handling (type2 keyword)	multiple-document- handling-default (type2 keyword)	multiple-document- handling-supported (1setOf type2 keyword)
2643			
2644			
2645			
2646	copies (integer (1:MAX))	copies-default (integer (1:MAX))	copies-supported (rangeOfInteger (1:MAX))
2647			
2648			
2649			
2650	finishings (1setOf type2 enum)	finishings-default (1setOf type2 enum)	finishings-supported (1setOf type2 enum)
2651			
2652			
2653	page-ranges (1setOf rangeOfInteger (1:MAX))	No	page-ranges- supported (boolean)
2654			
2655			
2656			
2657			
2658	sides (type2 keyword)	sides-default (type2 keyword)	sides-supported (1setOf type2 keyword)
2659			
2660			
2661	number-up (integer (1:MAX))	number-up-default (integer (1:MAX))	number-up-supported (1setOf integer (1:MAX) rangeOfInteger (1:MAX))
2662			
2663			
2664			
2665			
2666			
2667	orientation- requested (type2 enum)	orientation-requested- default (type2 enum)	orientation-requested- supported (1setOf type2 enum)
2668			
2669			
2670			
2671	media (type3 keyword name)	media-default (type3 keyword name)	media-supported (1setOf type3 keyword name)
2672			
2673			
2674			
2675			media-ready

2676			(1setOf
2677			type3 keyword name)
2678	+-----+-----+-----+		
2679	printer-resolution	printer-resolution-	printer-resolution-
2680	(resolution)	default	supported
2681		(resolution)	(1setOf resolution)
2682	+-----+-----+-----+		
2683	print-quality	print-quality-default	print-quality-
2684	(type2 enum)	(type2 enum)	supported
2685			(1setOf type2 enum)
2686	+-----+-----+-----+		

2687

2688

2689 4.2.1 job-priority (integer(1:100))

2690 This attribute specifies a priority for scheduling the Job. A higher value specifies a higher priority. The
 2691 value 1 indicates the lowest possible priority. The value 100 indicates the highest possible priority.
 2692 Among those jobs that are ready to print, a Printer MUST print all jobs with a priority value of n before
 2693 printing those with a priority value of n-1 for all n.

2694 If the Printer object supports this attribute, it MUST always support the full range from 1 to 100. No
 2695 administrative restrictions are permitted. This way an end-user can always make full use of the entire
 2696 range with any Printer object. If privileged jobs are implemented outside IPP/1.1, they MUST have
 2697 priorities higher than 100, rather than restricting the range available to end-users.

2698 If the client does not supply this attribute and this attribute is supported by the Printer object, the Printer
 2699 object MUST use the value of the Printer object's "job-priority-default" at job submission time (unlike
 2700 most Job Template attributes that are used if necessary at job processing time).

2701 The syntax for the "job-priority-supported" is also integer(1:100). This single integer value indicates the
 2702 number of priority levels supported. The Printer object MUST take the value supplied by the client and
 2703 map it to the closest integer in a sequence of n integers values that are evenly distributed over the range
 2704 from 1 to 100 using the formula:

2705
$$\text{roundToNearestInt}((100x+50)/n)$$

2706 where n is the value of "job-priority-supported" and x ranges from 0 through n-1.

2707 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,
 2708 the sequence of values is: 17, 50 and 83; if n = 10, the sequence of values is: 5, 15, 25, 35, 45, 55, 65,
 2709 75, 85, and 95; if n = 100, the sequence of values is: 1, 2, 3, ... 100.

2710 If the value of the Printer object's "job-priority-supported" is 10 and the client supplies values in the
 2711 range 1 to 10, the Printer object maps them to 5, in the range 11 to 20, the Printer object maps them to
 2712 15, etc.

2713 4.2.2 job-hold-until (type3 keyword | name (MAX))

2714 This attribute specifies the named time period during which the Job MUST become a candidate for
2715 printing.

2716 Standard keyword values for named time periods are:

2717 'no-hold': immediately, if there are not other reasons to hold the job

2718 'indefinite': - the job is held indefinitely, until a client performs a Release-Job (section 3.3.6)

2719 'day-time': during the day

2720 'evening': evening

2721 'night': night

2722 'weekend': weekend

2723 'second-shift': second-shift (after close of business)

2724 'third-shift': third-shift (after midnight)

2725

2726 An administrator MUST associate allowable print times with a named time period (by means outside the
2727 scope of this IPP/1.1 document). An administrator is encouraged to pick names that suggest the type of
2728 time period. An administrator MAY define additional values using the 'name' or 'keyword' attribute
2729 syntax, depending on implementation.

2730 If the value of this attribute specifies a time period that is in the future, the Printer MUST add the 'job-
2731 hold-until-specified' value to the job's "job-state-reasons" attribute, move the job to the 'pending-held'
2732 state, and MUST NOT schedule the job for printing until the specified time-period arrives. When the
2733 specified time period arrives, the Printer MUST remove the 'job-hold-until-specified' value from the
2734 job's "job-state-reason" attribute and, if there are no other job state reasons that keep the job in the
2735 'pending-held' state, the Printer MUST consider the job as a candidate for processing by moving the job
2736 to the 'pending' state.

2737 If this job attribute value is the named value 'no-hold', or the specified time period has already started,
2738 the job MUST be a candidate for processing immediately.

2739 If the client does not supply this attribute and this attribute is supported by the Printer object, the Printer
2740 object MUST use the value of the Printer object's "job-hold-until-default" at job submission time (unlike
2741 most Job Template attributes that are used if necessary at job processing time).

2742 4.2.3 job-sheets (type3 keyword | name(MAX))

2743 This attribute determines which job start/end sheet(s), if any, MUST be printed with a job.

2744 Standard keyword values are:

2745 'none': no job sheet is printed

2746 'standard': one or more site specific standard job sheets are printed, e.g. a single start sheet or both
2747 start and end sheet is printed

2748

2749 An administrator MAY define additional values using the 'name' or 'keyword' attribute syntax, depending
2750 on implementation.

2751 Note: The effect of this attribute on jobs with multiple documents MAY be affected by the "multiple-
2752 document-handling" job attribute (section 4.2.4), depending on the job sheet semantics.

2753 4.2.4 multiple-document-handling (type2 keyword)

2754 This attribute is relevant only if a job consists of two or more documents. The attribute controls finishing
2755 operations and the placement of one or more print-stream pages into impressions and onto media sheets.
2756 When the value of the "copies" attribute exceeds 1, it also controls the order in which the copies that
2757 result from processing the documents are produced. For the purposes of this explanations, if "a"
2758 represents an instance of document data, then the result of processing the data in document "a" is a
2759 sequence of media sheets represented by "a(*)".

2760 Standard keyword values are:

2761 'single-document': If a Job object has multiple documents, say, the document data is called a and b,
2762 then the result of processing all the document data (a and then b) MUST be treated as a single
2763 sequence of media sheets for finishing operations; that is, finishing would be performed on the
2764 concatenation of the sequences a(*),b(*). The Printer object MUST NOT force the data in each
2765 document instance to be formatted onto a new print-stream page, nor to start a new impression
2766 on a new media sheet. If more than one copy is made, the ordering of the sets of media sheets
2767 resulting from processing the document data MUST be a(*), b(*), a(*), b(*), ..., and the Printer
2768 object MUST force each copy (a(*),b(*)) to start on a new media sheet.

2769 'separate-documents-uncollated-copies': If a Job object has multiple documents, say, the document
2770 data is called a and b, then the result of processing the data in each document instance MUST be
2771 treated as a single sequence of media sheets for finishing operations; that is, the sets a(*) and b(*)
2772 would each be finished separately. The Printer object MUST force each copy of the result of
2773 processing the data in a single document to start on a new media sheet. If more than one copy is
2774 made, the ordering of the sets of media sheets resulting from processing the document data
2775 MUST be a(*), a(*), ..., b(*), b(*)

2776 'separate-documents-collated-copies': If a Job object has multiple documents, say, the document data
2777 is called a and b, then the result of processing the data in each document instance MUST be
2778 treated as a single sequence of media sheets for finishing operations; that is, the sets a(*) and b(*)
2779 would each be finished separately. The Printer object MUST force each copy of the result of
2780 processing the data in a single document to start on a new media sheet. If more than one copy is
2781 made, the ordering of the sets of media sheets resulting from processing the document data
2782 MUST be a(*), b(*), a(*), b(*),

2783 'single-document-new-sheet': Same as 'single-document', except that the Printer object MUST
2784 ensure that the first impression of each document instance in the job is placed on a new media
2785 sheet. This value allows multiple documents to be stapled together with a single staple where
2786 each document starts on a new sheet.
2787

2788 The 'single-document' value is the same as 'separate-documents-collated-copies' with respect to ordering
2789 of print-stream pages, but not media sheet generation, since 'single-document' will put the first page of

2790 the next document on the back side of a sheet if an odd number of pages have been produced so far for
 2791 the job, while 'separate-documents-collated-copies' always forces the next document or document copy
 2792 on to a new sheet. In addition, if the "finishings" attribute specifies 'staple', then with 'single-document',
 2793 documents a and b are stapled together as a single document with no regard to new sheets, with 'single-
 2794 document-new-sheet', documents a and b are stapled together as a single document, but document b
 2795 starts on a new sheet, but with 'separate-documents-uncollated-copies' and 'separate-documents-collated-
 2796 copies', documents a and b are stapled separately.

2797 Note: None of these values provide means to produce uncollated sheets within a document, i.e., where
 2798 multiple copies of sheet n are produced before sheet n+1 of the same document.

2799 The relationship of this attribute and the other attributes that control document processing is described in
 2800 section 16.3.

2801 4.2.5 copies (integer(1:MAX))

2802 This attribute specifies the number of copies to be printed.

2803 On many devices the supported number of collated copies will be limited by the number of physical
 2804 output bins on the device, and may be different from the number of uncollated copies which can be
 2805 supported.

2806 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-
 2807 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other
 2808 attributes that control document processing is described in section 16.3.

2809 4.2.6 finishings (1setOf type2 enum)

2810 This attribute identifies the finishing operations that the Printer uses for each copy of each printed
 2811 document in the Job. For Jobs with multiple documents, the "multiple-document-handling" attribute
 2812 determines what constitutes a "copy" for purposes of finishing.

2813 Standard enum values are:

2814	Value	Symbolic Name and Description
2815		
2816	'3'	'none': Perform no finishing
2817	'4'	'staple': Bind the document(s) with one or more staples. The exact number and placement 2818 of the staples is site-defined.
2819	'5'	'punch': This value indicates that holes are required in the finished document. The exact 2820 number and placement of the holes is site-defined The punch specification MAY 2821 be satisfied (in a site- and implementation-specific manner) either by 2822 drilling/punching, or by substituting pre-drilled media.
2823	'6'	'cover': This value is specified when it is desired to select a non-printed (or pre-printed) 2824 cover for the document. This does not supplant the specification of a printed cover 2825 (on cover stock medium) by the document itself.

- 2826 7' 'bind': This value indicates that a binding is to be applied to the document; the type and
2827 placement of the binding is site-defined.
2828
- 2829 8' 'saddle-stitch': Bind the document(s) with one or more staples (wire stitches) along the
2830 middle fold. The exact number and placement of the staples and the middle fold
2831 is implementation and/or site-defined.
- 2832 9' 'edge-stitch': Bind the document(s) with one or more staples (wire stitches) along one
2833 edge. The exact number and placement of the staples is implementation and/or
2834 site-defined.
- 2835 '10'-'19' reserved for future generic finishing enum values.

2836 The following values are more specific; they indicate a corner or an edge as if the document were a
2837 portrait document (see below):

- 2838 20' 'staple-top-left': Bind the document(s) with one or more staples in the top left corner.
2839 21' 'staple-bottom-left': Bind the document(s) with one or more staples in the bottom left
2840 corner.
2841 22' 'staple-top-right': Bind the document(s) with one or more staples in the top right corner.
2842 23' 'staple-bottom-right': Bind the document(s) with one or more staples in the bottom right
2843 corner.
2844 24' 'edge-stitch-left': Bind the document(s) with one or more staples (wire stitches) along the
2845 left edge. The exact number and placement of the staples is implementation
2846 and/or site-defined.
2847 25' 'edge-stitch-top': Bind the document(s) with one or more staples (wire stitches) along the
2848 top edge. The exact number and placement of the staples is implementation
2849 and/or site-defined.
2850 26' 'edge-stitch-right': Bind the document(s) with one or more staples (wire stitches) along
2851 the right edge. The exact number and placement of the staples is implementation
2852 and/or site-defined.
2853 27' 'edge-stitch-bottom': Bind the document(s) with one or more staples (wire stitches) along
2854 the bottom edge. The exact number and placement of the staples is
2855 implementation and/or site-defined.
2856 28' 'staple-dual-left': Bind the document(s) with two staples (wire stitches) along the left
2857 edge assuming a portrait document (see above).
2858 29' 'staple-dual-top': Bind the document(s) with two staples (wire stitches) along the top
2859 edge assuming a portrait document (see above).
2860 30' 'staple-dual-right': Bind the document(s) with two staples (wire stitches) along the right
2861 edge assuming a portrait document (see above).
2862 31' 'staple-dual-bottom': Bind the document(s) with two staples (wire stitches) along the
2863 bottom edge assuming a portrait document (see above).

2864 The 'staple-xxx' values are specified with respect to the document as if the document were a portrait
2865 document. If the document is actually a landscape or a reverse-landscape document, the client supplies
2866 the appropriate transformed value. For example, to position a staple in the upper left hand corner of a
2867 landscape document when held for reading, the client supplies the 'staple-bottom-left' value (since
2868 landscape is defined as a +90 degree rotation from portrait, i.e., anti-clockwise). On the other hand, to
2869 position a staple in the upper left hand corner of a reverse-landscape document when held for reading,

2870 the client supplies the 'staple-top-right' value (since reverse-landscape is defined as a -90 degree rotation
2871 from portrait, i.e., clockwise).

2872 The angle (vertical, horizontal, angled) of each staple with respect to the document depends on the
2873 implementation which may in turn depend on the value of the attribute.

2874 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-
2875 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other
2876 attributes that control document processing is described in section 16.3.

2877 If the client supplies a value of 'none' along with any other combination of values, it is the same as if
2878 only that other combination of values had been supplied (that is the 'none' value has no effect).

2879 4.2.7 page-ranges (1setOf rangeOfInteger (1:MAX))

2880 This attribute identifies the range(s) of print-stream pages that the Printer object uses for each copy of
2881 each document which are to be printed. Nothing is printed for any pages identified that do not exist in
2882 the document(s). Ranges MUST be in ascending order, for example: 1-3, 5-7, 15-19 and MUST NOT
2883 overlap, so that a non-spooling Printer object can process the job in a single pass. If the ranges are not
2884 ascending or are overlapping, the IPP object MUST reject the request and return the 'client-error-bad-
2885 request' status code. The attribute is associated with print-stream pages not application-numbered pages
2886 (for example, the page numbers found in the headers and or footers for certain word processing
2887 applications).

2888 For Jobs with multiple documents, the "multiple-document-handling" attribute determines what
2889 constitutes a "copy" for purposes of the specified page range(s). When "multiple-document-handling" is
2890 'single-document', the Printer object MUST apply each supplied page range once to the concatenation of
2891 the print-stream pages. For example, if there are 8 documents of 10 pages each, the page-range '41:60'
2892 prints the pages in the 5th and 6th documents as a single document and none of the pages of the other
2893 documents are printed. When "multiple-document-handling" is 'separate-document-uncollated-copies' or
2894 'separate-document-collated-copies', the Printer object MUST apply each supplied page range repeatedly
2895 to each document copy. For the same job, the page-range '1:3, 10:10' would print the first 3 pages and
2896 the 10th page of each of the 8 documents in the Job, as 8 separate documents.

2897 In most cases, the exact pages to be printed will be generated by a device driver and this attribute would
2898 not be required. However, when printing an archived document which has already been formatted, the
2899 end user may elect to print just a subset of the pages contained in the document. In this case, if page-
2900 range = n.m is specified, the first page to be printed will be page n. All subsequent pages of the
2901 document will be printed through and including page m.

2902 "page-ranges-supported" is a boolean value indicating whether or not the printer is capable of supporting
2903 the printing of page ranges. This capability may differ from one PDL to another. There is no "page-
2904 ranges-default" attribute. If the "page-ranges" attribute is not supplied by the client, all pages of the
2905 document will be printed.

2906 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-
 2907 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other
 2908 attributes that control document processing is described in section 16.3.

2909 4.2.8 sides (type2 keyword)

2910 This attribute specifies how print-stream pages are to be imposed upon the sides of an instance of a
 2911 selected medium, i.e., an impression.

2912 The standard keyword values are:

2913 'one-sided': imposes each consecutive print-stream page upon the same side of consecutive media
 2914 sheets.

2915 'two-sided-long-edge': imposes each consecutive pair of print-stream pages upon front and back sides
 2916 of consecutive media sheets, such that the orientation of each pair of print-stream pages on the
 2917 medium would be correct for the reader as if for binding on the long edge. This imposition is
 2918 sometimes called 'duplex' or 'head-to-head'.

2919 'two-sided-short-edge': imposes each consecutive pair of print-stream pages upon front and back
 2920 sides of consecutive media sheets, such that the orientation of each pair of print-stream pages on
 2921 the medium would be correct for the reader as if for binding on the short edge. This imposition
 2922 is sometimes called 'tumble' or 'head-to-toe'.

2923

2924 'two-sided-long-edge', 'two-sided-short-edge', 'tumble', and 'duplex' all work the same for portrait or
 2925 landscape. However 'head-to-toe' is 'tumble' in portrait but 'duplex' in landscape. 'head-to-head' also
 2926 switches between 'duplex' and 'tumble' when using portrait and landscape modes.

2927 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-
 2928 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other
 2929 attributes that control document processing is described in section 16.3.

2930 4.2.9 number-up (integer(1:MAX))

2931 This attribute specifies the number of print-stream pages to impose upon a single side of an instance of a
 2932 selected medium. For example, if the value is:

2933 Value	Description
2934 '1'	the Printer MUST place one print-stream page on a single side of an instance of the 2935 selected medium (MAY add some sort of translation, scaling, or rotation).
2936 '2'	the Printer MUST place two print-stream pages on a single side of an instance of the 2937 selected medium (MAY add some sort of translation, scaling, or rotation).
2938 '4'	the Printer MUST place four print-stream pages on a single side of an instance of the 2939 selected medium (MAY add some sort of translation, scaling, or rotation).
2940	
2941	

2941

2942 This attribute primarily controls the translation, scaling and rotation of print-stream pages.

2943 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-
 2944 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other
 2945 attributes that control document processing is described in section 16.3.

2946 4.2.10 orientation-requested (type2 enum)

2947 This attribute indicates the desired orientation for printed print-stream pages; it does not describe the
 2948 orientation of the client-supplied print-stream pages.

2949 For some document formats (such as 'application/postscript'), the desired orientation of the print-stream
 2950 pages is specified within the document data. This information is generated by a device driver prior to
 2951 the submission of the print job. Other document formats (such as 'text/plain') do not include the notion
 2952 of desired orientation within the document data. In the latter case it is possible for the Printer object to
 2953 bind the desired orientation to the document data after it has been submitted. It is expected that a Printer
 2954 object would only support "orientations-requested" for some document formats (e.g., 'text/plain' or
 2955 'text/html') but not others (e.g., 'application/postscript'). This is no different than any other Job Template
 2956 attribute since section 4.2, item 1, points out that a Printer object may support or not support any Job
 2957 Template attribute based on the document format supplied by the client. However, a special mention is
 2958 made here since it is very likely that a Printer object will support "orientation-requested" for only a
 2959 subset of the supported document formats.

2960 Standard enum values are:

2961	Value	Symbolic Name and Description
2962		
2963	'3'	'portrait': The content will be imaged across the short edge of the medium.
2964	'4'	'landscape': The content will be imaged across the long edge of the medium. Landscape
2965		is defined to be a rotation of the print-stream page to be imaged by +90 degrees
2966		with respect to the medium (i.e. anti-clockwise) from the portrait orientation.
2967		Note: The +90 direction was chosen because simple finishing on the long edge is
2968		the same edge whether portrait or landscape
2969	'5'	'reverse-landscape': The content will be imaged across the long edge of the medium.
2970		Reverse-landscape is defined to be a rotation of the print-stream page to be
2971		imaged by -90 degrees with respect to the medium (i.e. clockwise) from the
2972		portrait orientation. Note: The 'reverse-landscape' value was added because some
2973		applications rotate landscape -90 degrees from portrait, rather than +90 degrees.
2974	'6'	'reverse-portrait': The content will be imaged across the short edge of the medium.
2975		Reverse-portrait is defined to be a rotation of the print-stream page to be imaged
2976		by 180 degrees with respect to the medium from the portrait orientation. Note:
2977		The 'reverse-portrait' value was added for use with the "finishings" attribute in
2978		cases where the opposite edge is desired for finishing a portrait document on
2979		simple finishing devices that have only one finishing position. Thus a 'text/plain'
2980		portrait document can be stapled "on the right" by a simple finishing device as is
2981		common use with some middle eastern languages such as Hebrew.
2982		

2983 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-
2984 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other
2985 attributes that control document processing is described in section 16.3.

2986 4.2.11 media (type3 keyword | name(MAX))

2987 This attribute identifies the medium that the Printer uses for all impressions of the Job.

2988 The values for "media" include medium-names, medium-sizes, input-trays and electronic forms so that
2989 one attribute specifies the media. If a Printer object supports a medium name as a value of this attribute,
2990 such a medium name implicitly selects an input-tray that contains the specified medium. If a Printer
2991 object supports a medium size as a value of this attribute, such a medium size implicitly selects a
2992 medium name that in turn implicitly selects an input-tray that contains the medium with the specified
2993 size. If a Printer object supports an input-tray as the value of this attribute, such an input-tray implicitly
2994 selects the medium that is in that input-tray at the time the job prints. This case includes manual-feed
2995 input-trays. If a Printer object supports an electronic form as the value of this attribute, such an
2996 electronic form implicitly selects a medium-name that in turn implicitly selects an input-tray that
2997 contains the medium specified by the electronic form. The electronic form also implicitly selects an
2998 image that the Printer MUST merge with the document data as its prints each page.

2999 Standard keyword values are (taken from ISO DPA and the Printer MIB) and are listed in section 15. An
3000 administrator MAY define additional values using the 'name' or 'keyword' attribute syntax, depending on
3001 implementation.

3002 There is also an additional Printer attribute named "media-ready" which differs from "media-supported"
3003 in that legal values only include the subset of "media-supported" values that are physically loaded and
3004 ready for printing with no operator intervention required. If an IPP object supports "media-supported", it
3005 NEED NOT support "media-ready".

3006 The relationship of this attribute and the other attributes that control document processing is described in
3007 section 16.3.

3008 4.2.12 printer-resolution (resolution)

3009 This attribute identifies the resolution that Printer uses for the Job.

3010 4.2.13 print-quality (type2 enum)

3011 This attribute specifies the print quality that the Printer uses for the Job.

3012 The standard enum values are:

3013	Value	Symbolic Name and Description
3014		
3015	'3'	'draft': lowest quality available on the printer
3016	'4'	'normal': normal or intermediate quality on the printer

3017 '5' 'high': highest quality available on the printer
3018

3019 4.3 Job Description Attributes

3020 The attributes in this section form the attribute group called "job-description". The following table
3021 summarizes these attributes. The third column indicates whether the attribute is a REQUIRED attribute
3022 that MUST be supported by Printer objects. If it is not indicated as REQUIRED, then it is OPTIONAL.
3023 The maximum size in octets for 'text' and 'name' attributes is indicated in parentheses.

3024	+	-----+	+	-----+	+	-----+
3025		Attribute		Syntax		REQUIRED?
3026	+	-----+	+	-----+	+	-----+
3027		job-uri		uri		REQUIRED
3028	+	-----+	+	-----+	+	-----+
3029		job-id		integer(1:MAX)		REQUIRED
3030	+	-----+	+	-----+	+	-----+
3031		job-printer-uri		uri		REQUIRED
3032	+	-----+	+	-----+	+	-----+
3033		job-more-info		uri		
3034	+	-----+	+	-----+	+	-----+
3035		job-name		name (MAX)		REQUIRED
3036	+	-----+	+	-----+	+	-----+
3037		job-originating-user-name		name (MAX)		REQUIRED
3038	+	-----+	+	-----+	+	-----+
3039		job-state		type1 enum		REQUIRED
3040	+	-----+	+	-----+	+	-----+
3041		job-state-reasons		1setOf type2 keyword		
3042	+	-----+	+	-----+	+	-----+
3043		job-state-message		text (MAX)		
3044	+	-----+	+	-----+	+	-----+
3045		number-of-documents		integer (0:MAX)		
3046	+	-----+	+	-----+	+	-----+
3047		output-device-assigned		name (127)		
3048	+	-----+	+	-----+	+	-----+
3049		time-at-creation		integer (0:MAX)		
3050	+	-----+	+	-----+	+	-----+
3051		time-at-processing		integer (0:MAX)		
3052	+	-----+	+	-----+	+	-----+
3053		time-at-completed		integer (0:MAX)		
3054	+	-----+	+	-----+	+	-----+
3055		number-of-intervening-jobs		integer (0:MAX)		
3056	+	-----+	+	-----+	+	-----+
3057		job-message-from-operator		text (127)		
3058	+	-----+	+	-----+	+	-----+
3059		job-k-octets		integer (0:MAX)		
3060	+	-----+	+	-----+	+	-----+
3061		job-impressions		integer (0:MAX)		
3062	+	-----+	+	-----+	+	-----+
3063		job-media-sheets		integer (0:MAX)		
3064	+	-----+	+	-----+	+	-----+
3065		job-k-octets-processed		integer (0:MAX)		
3066	+	-----+	+	-----+	+	-----+
3067		job-impressions-completed		integer (0:MAX)		
3068	+	-----+	+	-----+	+	-----+
3069		job-media-sheets-completed		integer (0:MAX)		
3070	+	-----+	+	-----+	+	-----+
3071		attributes-charset		charset		REQUIRED
3072	+	-----+	+	-----+	+	-----+
3073		attributes-natural-language		naturalLanguage		REQUIRED

3074 +-----+-----+-----+-----+
3075
3076

3077 4.3.1 job-uri (uri)

3078 This REQUIRED attribute contains the URI for the job. The Printer object, on receipt of a new job,
3079 generates a URI which identifies the new Job. The Printer object returns the value of the "job-uri"
3080 attribute as part of the response to a create request. The precise format of a Job URI is implementation
3081 dependent. If the Printer object supports more than one URI and there is some relationship between the
3082 newly formed Job URI and the Printer object's URI, the Printer object uses the Printer URI supplied by
3083 the client in the create request. For example, if the create request comes in over a secure channel, the
3084 new Job URI MUST use the same secure channel. This can be guaranteed because the Printer object is
3085 responsible for generating the Job URI and the Printer object is aware of its security configuration and
3086 policy as well as the Printer URI used in the create request.

3087 For a description of this attribute and its relationship to "job-id" and "job-printer-uri" attribute, see the
3088 discussion in section 2.4 on "Object Identity".

3089 4.3.2 job-id (integer(1:MAX))

3090 This REQUIRED attribute contains the ID of the job. The Printer, on receipt of a new job, generates an
3091 ID which identifies the new Job on that Printer. The Printer returns the value of the "job-id" attribute as
3092 part of the response to a create request. The 0 value is not included to allow for compatibility with
3093 SNMP index values which also cannot be 0.

3094 For a description of this attribute and its relationship to "job-uri" and "job-printer-uri" attribute, see the
3095 discussion in section 2.4 on "Object Identity".

3096 4.3.3 job-printer-uri (uri)

3097 This REQUIRED attribute identifies the Printer object that created this Job object. When a Printer
3098 object creates a Job object, it populates this attribute with the Printer object URI that was used in the
3099 create request. This attribute permits a client to identify the Printer object that created this Job object
3100 when only the Job object's URI is available to the client. The client queries the creating Printer object to
3101 determine which languages, charsets, operations, are supported for this Job.

3102 For a description of this attribute and its relationship to "job-uri" and "job-id" attribute, see the
3103 discussion in section 2.4 on "Object Identity".

3104 4.3.4 job-more-info (uri)

3105 Similar to "printer-more-info", this attribute contains the URI referencing some resource with more
3106 information about this Job object, perhaps an HTML page containing information about the Job.

3107 4.3.5 job-name (name(MAX))

3108 This REQUIRED attribute is the name of the job. It is a name that is more user friendly than the "job-
 3109 uri" attribute value. It does not need to be unique between Jobs. The Job's "job-name" attribute is set to
 3110 the value supplied by the client in the "job-name" operation attribute in the create request (see Section
 3111 3.2.1.1). If, however, the "job-name" operation attribute is not supplied by the client in the create
 3112 request, the Printer object, on creation of the Job, MUST generate a name. The printer SHOULD
 3113 generate the value of the Job's "job-name" attribute from the first of the following sources that produces
 3114 a value: 1) the "document-name" operation attribute of the first (or only) document, 2) the "document-
 3115 URI" attribute of the first (or only) document, or 3) any other piece of Job specific and/or Document
 3116 Content information.

3117 4.3.6 job-originating-user-name (name(MAX))

3118 This REQUIRED attribute contains the name of the end user that submitted the print job. The Printer
 3119 object sets this attribute to the most authenticated printable name that it can obtain from the
 3120 authentication service over which the IPP operation was received. Only if such is not available, does the
 3121 Printer object use the value supplied by the client in the "requesting-user-name" operation attribute of the
 3122 create operation (see Section 8).

3123 Note: The Printer object needs to keep an internal originating user id of some form, typically as a
 3124 credential of a principal, with the Job object. Since such an internal attribute is implementation-
 3125 dependent and not of interest to clients, it is not specified as a Job Description attribute. This originating
 3126 user id is used for authorization checks (if any) on all subsequent operation.

3127 4.3.7 job-state (type1 enum)

3128 This REQUIRED attribute identifies the current state of the job. Even though the IPP protocol defines
 3129 seven values for job states (plus the out-of-band 'unknown' value - see Section 4.1), implementations
 3130 only need to support those states which are appropriate for the particular implementation. In other
 3131 words, a Printer supports only those job states implemented by the output device and available to the
 3132 Printer object implementation.

3133 Standard enum values are:

3134	Values	Symbolic Name and Description
3135		
3136	'3'	'pending': The job is a candidate to start processing, but is not yet processing.
3137		
3138	'4'	'pending-held': The job is not a candidate for processing for any number of reasons but will return to the 'pending' state as soon as the reasons are no longer present. The job's "job-state-reason" attribute MUST indicate why the job is no longer a candidate for processing.
3139		
3140		
3141		
3142		
3143	'5'	'processing': One or more of:
3144		

- 3145 1. the job is using, or is attempting to use, one or more purely software processes
3146 that are analyzing, creating, or interpreting a PDL, etc.,
3147 2. the job is using, or is attempting to use, one or more hardware devices that are
3148 interpreting a PDL, making marks on a medium, and/or performing finishing,
3149 such as stapling, etc.,
3150 3. the Printer object has made the job ready for printing, but the output device is
3151 not yet printing it, either because the job hasn't reached the output device or
3152 because the job is queued in the output device or some other spooler, awaiting the
3153 output device to print it.
3154

3155 When the job is in the 'processing' state, the entire job state includes the detailed
3156 status represented in the printer's "printer-state", "printer-state-reasons", and
3157 "printer-state-message" attributes.

3158 Implementations MAY, though they NEED NOT, include additional values in the
3159 job's "job-state-reasons" attribute to indicate the progress of the job, such as
3160 adding the 'job-printing' value to indicate when the output device is actually
3161 making marks on paper and/or the 'processing-to-stop-point' value to indicate that
3162 the IPP object is in the process of canceling or aborting the job. Most
3163 implementations won't bother with this nuance.

3164
3165 '6' 'processing-stopped': The job has stopped while processing for any number of reasons
3166 and will return to the 'processing' state as soon as the reasons are no longer
3167 present.
3168

3169 The job's "job-state-reason" attribute MAY indicate why the job has stopped
3170 processing. For example, if the output device is stopped, the 'printer-stopped'
3171 value MAY be included in the job's "job-state-reasons" attribute.

3172
3173 Note: When an output device is stopped, the device usually indicates its condition
3174 in human readable form locally at the device. A client can obtain more complete
3175 device status remotely by querying the Printer object's "printer-state", "printer-
3176 state-reasons" and "printer-state-message" attributes.

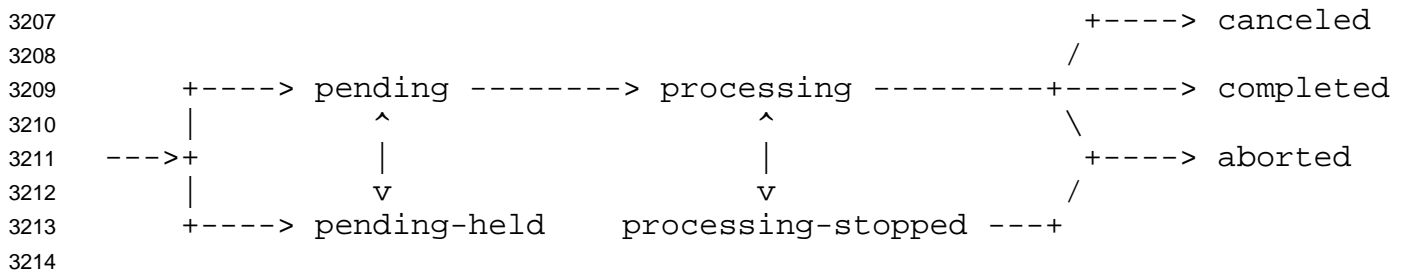
3177
3178 '7' 'canceled': The job has been canceled by a Cancel-Job operation and the Printer object
3179 has completed canceling the job and all job status attributes have reached their
3180 final values for the job. While the Printer object is canceling the job, the job
3181 remains in its current state, but the job's "job-state-reasons" attribute SHOULD
3182 contain the 'processing-to-stop-point' value and one of the 'canceled-by-user',
3183 'canceled-by-operator', or 'canceled-at-device' value. When the job moves to the
3184 'canceled' state, the 'processing-to-stop-point' value, if present, MUST be
3185 removed, but the 'canceled-by-xxx', if present, MUST remain.
3186

3187 '8' 'aborted': The job has been aborted by the system, usually while the job was in the
 3188 'processing' or 'processing-stopped' state and the Printer has completed aborting
 3189 the job and all job status attributes have reached their final values for the job.
 3190 While the Printer object is aborting the job, the job remains in its current state, but
 3191 the job's "job-state-reasons" attribute SHOULD contain the 'processing-to-stop-
 3192 point' and 'aborted-by-system' values. When the job moves to the 'aborted' state,
 3193 the 'processing-to-stop-point' value, if present, MUST be removed, but the
 3194 'aborted-by-system' value, if present, MUST remain.

3196 '9' 'completed': The job has completed successfully or with warnings or errors after
 3197 processing and all of the job media sheets have been successfully stacked in the
 3198 appropriate output bin(s) and all job status attributes have reached their final
 3199 values for the job. The job's "job-state-reasons" attribute SHOULD contain one
 3200 of: 'completed-successfully', 'completed-with-warnings', or 'completed-with-errors'
 3201 values.
 3202

3203 The final value for this attribute MUST be one of: 'completed', 'canceled', or 'aborted' before the Printer
 3204 removes the job altogether. The length of time that jobs remain in the 'canceled', 'aborted', and
 3205 'completed' states depends on implementation. See section 4.3.7.1.

3206 The following figure shows the normal job state transitions.



3215 Normally a job progresses from left to right. Other state transitions are unlikely, but are not forbidden.
 3216 Not shown are the transitions to the 'canceled' state from the 'pending', 'pending-held', and 'processing-
 3217 stopped' states.

3218 Jobs reach one of the three terminal states: 'completed', 'canceled', or 'aborted', after the jobs have
 3219 completed all activity, including stacking output media, after the jobs have completed all activity, and all
 3220 job status attributes have reached their final values for the job.

3221 Note: As with all other IPP attributes, if the implementation can not determine the correct value for this
 3222 attribute, it SHOULD respond with the out-of-band value 'unknown' (see section 4.1) rather than try to
 3223 guess at some possibly incorrect value and give the end user the wrong impression about the state of the
 3224 Job object. For example, if the implementation is just a gateway into some printing system that does not
 3225 provide detailed status about the print job, the IPP Job object's state might literally be 'unknown'.

3226 4.3.7.1 Partitioning of Job States

3227 This section partitions the 7 job states into phases: Job Not Completed, Job Retention, Job History, and
3228 Job Removal. This section also explains the 'job-restartable' value of the "job-state-reasons" Job
3229 Description attribute for use with the Restart-Job operation.

3230 Job Not Completed: When a job is in the 'pending', 'pending-held', 'processing', or 'processing-stopped'
3231 states, the job is not completed.

3232 Job Retention: When a job enters one of the three terminal job states: 'completed', 'canceled', or
3233 'aborted', the IPP Printer object MAY "retain" the job in a restartable condition for an implementation-
3234 defined time period. This time period MAY be zero seconds and MAY depend on the terminal job state.
3235 This phase is called Job Retention. While in the Job Retention phase, the job's document data is retained
3236 and a client may restart the job using the Restart-Job operation. If the IPP object supports the "job-state-
3237 reasons" attribute and the Restart-Job operation, then it SHOULD indicate that the job is restartable by
3238 adding the 'job-restartable' value to the job's "job-state-reasons" attribute (see Section 4.3.8) during the
3239 Job Retention phase.

3240 Job History: After the Job Retention phase expires for a job, the Printer object deletes the document
3241 data for the job and the job becomes part of the Job History. The Printer object MAY also delete any
3242 number of the job attributes. Since the job is no longer restartable, the Printer object MUST remove the
3243 'job-restartable' value from the job's "job-state-reasons" attribute, if present.

3244 Job Removal: After the job has remained in the Job History for an implementation-defined time, such as
3245 when the number of jobs exceeds a fixed number or after a fixed time period (which MAY be zero
3246 seconds), the IPP Printer removes the job from the system.

3247 Using the Get-Jobs operation and supplying the 'not-completed' value for the "which-jobs" operation
3248 attribute, a client is requesting jobs in the Job Not Completed phase. Using the Get-Jobs operation and
3249 supplying the 'completed' value for the "which-jobs" operation attribute, a client is requesting jobs in the
3250 Job Retention and Job History phases. Using the Get-Job-Attributes operation, a client is requesting a
3251 job in any phase except Job Removal. After Job Removal, the Get-Job-Attributes and Get-Jobs
3252 operations no longer are capable of returning any information about a job.

3253 4.3.8 job-state-reasons (1setOf type2 keyword)

3254 This attribute provides additional information about the job's current state, i.e., information that
3255 augments the value of the job's "job-state" attribute.

3256 Implementation of these values is OPTIONAL, i.e., a Printer NEED NOT implement them, even if (1)
3257 the output device supports the functionality represented by the reason and (2) is available to the Printer
3258 object implementation. These values MAY be used with any job state or states for which the reason
3259 makes sense. Furthermore, when implemented, the Printer MUST return these values when the reason
3260 applies and MUST NOT return them when the reason no longer applies whether the value of the Job's
3261 "job-state" attribute changed or not. When the Job does not have any reasons for being in its current
3262 state, the value of the Job's "job-state-reasons" attribute MUST be 'none'.

3263 Note: While values cannot be added to the 'job-state' attribute without impacting deployed clients that
3264 take actions upon receiving "job-state" values, it is the intent that additional "job-state-reasons" values
3265 can be defined and registered without impacting such deployed clients. In other words, the "job-state-
3266 reasons" attribute is intended to be extensible.

3267 The following standard keyword values are defined. For ease of understanding, the values are presented
3268 in the order in which the reasons are likely to occur (if implemented), starting with the 'job-incoming'
3269 value:

3270 'none': There are no reasons for the job's current state.

3271 'job-incoming': The Create-Job operation has been accepted by the Printer, but the Printer is
3272 expecting additional Send-Document and/or Send-URI operations and/or is accessing/accepting
3273 document data.

3274 'submission-interrupted': The job was not completely submitted for some unforeseen reason, such
3275 as: (1) the Printer has crashed before the job was closed by the client, (2) the Printer or the
3276 document transfer method has crashed in some non-recoverable way before the document data
3277 was entirely transferred to the Printer, (3) the client crashed or failed to close the job before the
3278 time-out period. See section 4.4.28.

3279 'job-outgoing': The Printer is transmitting the job to the output device.

3280 'job-hold-until-specified': The value of the job's "job-hold-until" attribute was specified with a time
3281 period that is still in the future. The job MUST NOT be a candidate for processing until this
3282 reason is removed and there are no other reasons to hold the job.

3283 'resources-are-not-ready': At least one of the resources needed by the job, such as media, fonts,
3284 resource objects, etc., is not ready on any of the physical printer's for which the job is a candidate.
3285 This condition MAY be detected when the job is accepted, or subsequently while the job is
3286 pending or processing, depending on implementation. The job may remain in its current state or
3287 be moved to the 'pending-held' state, depending on implementation and/or job scheduling policy.

3288 'printer-stopped-partly': The value of the Printer's "printer-state-reasons" attribute contains the value
3289 'stopped-partly'.

3290 'printer-stopped': The value of the Printer's "printer-state" attribute is 'stopped'.

3291 'job-interpreting': Job is in the 'processing' state, but more specifically, the Printer is interpreting the
3292 document data.

3293 'job-queued': Job is in the 'processing' state, but more specifically, the Printer has queued the
3294 document data.

3295 'job-transforming': Job is in the 'processing' state, but more specifically, the Printer is interpreting
3296 document data and producing another electronic representation.

3297 'job-printing': The output device is marking media. This value is useful for Printers which spend a
3298 great deal of time processing (1) when no marking is happening and then want to show that
3299 marking is now happening or (2) when the job is in the process of being canceled or aborted
3300 while the job remains in the 'processing' state, but the marking has not yet stopped so that
3301 impression or sheet counts are still increasing for the job.

3302 'job-canceled-by-user': The job was canceled by the owner of the job using the Cancel-Job request,
3303 i.e., by a user whose authenticated identity is the same as the value of the originating user that
3304 created the Job object, or by some other authorized end-user, such as a member of the job
3305 owner's security group.

3306 'job-canceled-by-operator': The job was canceled by the operator using the Cancel-Job request, i.e.,
3307 by a user who has been authenticated as having operator privileges (whether local or remote). If
3308 the security policy is to allow anyone to cancel anyone's job, then this value may be used when
3309 the job is canceled by other than the owner of the job. For such a security policy, in effect,
3310 everyone is an operator as far as canceling jobs with IPP is concerned.

3311 'job-canceled-at-device': The job was canceled by an unidentified local user, i.e., a user at a console
3312 at the device.

3313 'aborted-by-system': The job (1) is in the process of being aborted, (2) has been aborted by the
3314 system and placed in the 'aborted' state, or (3) has been aborted by the system and placed in the
3315 'pending-held' state, so that a user or operator can manually try the job again.

3316 'unsupported-compression': The job was aborted by the system because the document-data's
3317 compression is not among those supported by the printer. ISSUE #6

3318 'compression-error': The job was aborted by the system because the printer encountered an error in
3319 the document-data while decompressing it. If the printer posts this reason, the document-data has
3320 already passed any tests that would have led to the 'unsupported-compression' job-state-reason.
3321 ISSUE #6

3322 'unsupported-document-format': The job was aborted by the system because the document-data's
3323 document-format is not among those supported by the printer. If the client specifies the
3324 document-format as 'application/octet-stream', the printer may abort the job and post this reason
3325 even though the format is a member of the "document-format-supported" printer attribute, but
3326 not among the auto-sensed document-formats. ISSUE #3

3327 'document-format-error': The job was aborted by the system because the printer encountered an error
3328 in the document-data while processing it. If the printer posts this reason, the document-data has
3329 already passed any tests that would have led to the 'unsupported-document-format' job-state-
3330 reason. ISSUE #3

3331 'processing-to-stop-point': The requester has issued a Cancel-job operation or the Printer object has
3332 aborted the job, but is still performing some actions on the job until a specified stop point occurs
3333 or job termination/cleanup is completed.

3334

3335 This reason is recommended to be used in conjunction with the 'processing' job state to indicate
3336 that the Printer object is still performing some actions on the job while the job remains in the
3337 'processing' state. After all the job's job description attributes have stopped incrementing, the
3338 Printer object moves the job from the 'processing' state to the 'canceled' or 'aborted' job states.

3339

3340 'service-off-line': The Printer is off-line and accepting no jobs. All 'pending' jobs are put into the
3341 'pending-held' state. This situation could be true if the service's or document transform's input is
3342 impaired or broken.

3343 'job-completed-successfully': The job completed successfully.

3344 'job-completed-with-warnings': The job completed with warnings.

3345 'job-completed-with-errors': The job completed with errors (and possibly warnings too).

3346 'job-restartable' - This job is retained (see section 4.3.7.1) and is currently able to be restarted using
3347 the Restart-Job operation (see section 3.3.7). If 'job-restartable' is a value of the job's 'job-state-

3348 reasons' attribute, then the IPP object MUST accept a Restart-Job operation for that job.

3349 'queued-in-device': The job has been forwarded to a Printer which is unable to send back status. The
3350 Printer sets the job's "job-state " to 'completed' and adds 'queued-in-device' to the "job-state-
3351 reasons" to indicate that the Printer has no additional information about the job and never will
3352 have any better information. Issue 14
3353

3354 4.3.9 job-state-message (text(MAX))

3355 This attribute specifies information about the "job-state" and "job-state-reasons" attributes in human
3356 readable text. If the Printer object supports this attribute, the Printer object MUST be able to generate
3357 this message in any of the natural languages identified by the Printer's "generated-natural-language-
3358 supported" attribute (see the "attributes-natural-language" operation attribute specified in Section
3359 3.1.4.1).

3360 Note: the value SHOULD NOT contain additional information not contained in the values of the "job-
3361 state" and "job-states-reasons" attributes, such as interpreter error information. Otherwise, application
3362 programs might attempt to parse the (localized text). For such additional information such as interpreter
3363 errors for application program consumption, a new attribute with keyword values, needs to be developed
3364 and registered.

3365 4.3.10 number-of-documents (integer(0:MAX))

3366 This attribute indicates the number of documents in the job, i.e., the number of Send-Document, Send-
3367 URI, Print-Job, or Print-URI operations that the Printer has accepted for this job, regardless of whether
3368 the document data has reached the Printer object or not.

3369 Implementations supporting the OPTIONAL Create-Job/Send-Document/Send-URI operations
3370 SHOULD support this attribute so that clients can query the number of documents in each job.

3371 4.3.11 output-device-assigned (name(127))

3372 This attribute identifies the output device to which the Printer object has assigned this job. If an output
3373 device implements an embedded Printer object, the Printer object NEED NOT set this attribute. If a
3374 print server implements a Printer object, the value MAY be empty (zero-length string) or not returned
3375 until the Printer object assigns an output device to the job. This attribute is particularly useful when a
3376 single Printer object support multiple devices (so called "fan-out").

3377 4.3.12 time-at-creation (integer(0:MAX))

3378 This attribute indicates the point in time at which the Job object was created. In order to populate this
3379 attribute, the Printer object uses the value in its "printer-up-time" attribute at the time the Job object is
3380 created.

3381 4.3.13 time-at-processing (integer(0:MAX))

3382 This attribute indicates the point in time at which the Job object began processing. In order to populate
3383 this attribute, the Printer object uses the value in its "printer-up-time" attribute at the time the Job object
3384 is moved into the 'processing' state for the first time.

3385 4.3.14 time-at-completed (integer(0:MAX))

3386 This attribute indicates the point in time at which the Job object completed (or was cancelled or aborted).
3387 In order to populate this attribute, the Printer object uses the value in its "printer-up-time" attribute at the
3388 time the Job object is moved into the 'completed' or 'canceled' or 'aborted' state.

3389 4.3.15 number-of-intervening-jobs (integer(0:MAX))

3390 This attribute indicates the number of jobs that are "ahead" of this job in the relative chronological order
3391 of expected time to complete (i.e., the current scheduled order). For efficiency, it is only necessary to
3392 calculate this value when an operation is performed that requests this attribute.

3393 4.3.16 job-message-from-operator (text(127))

3394 This attribute provides a message from an operator, system administrator or "intelligent" process to
3395 indicate to the end user the reasons for modification or other management action taken on a job.

3396 4.3.17 job-k-octets (integer(0:MAX))

3397 This attribute specifies the total size of the document(s) in K octets, i.e., in units of 1024 octets requested
3398 to be processed in the job. The value MUST be rounded up, so that a job between 1 and 1024 octets
3399 MUST be indicated as being 1, 1025 to 2048 MUST be 2, etc.

3400 This value MUST NOT include the multiplicative factors contributed by the number of copies specified
3401 by the "copies" attribute, independent of whether the device can process multiple copies without making
3402 multiple passes over the job or document data and independent of whether the output is collated or not.
3403 Thus the value is independent of the implementation and indicates the size of the document(s) measured
3404 in K octets independent of the number of copies.

3405 This value MUST also not include the multiplicative factor due to a copies instruction embedded in the
3406 document data. If the document data actually includes replications of the document data, this value will
3407 include such replication. In other words, this value is always the size of the source document data, rather
3408 than a measure of the hardcopy output to be produced.

3409 Note: This attribute and the following two attributes ("job-impressions" and "job-media-sheets") are not
3410 intended to be counters; they are intended to be useful routing and scheduling information if known. For
3411 these three attributes, the Printer object may try to compute the value if it is not supplied in the create
3412 request. Even if the client does supply a value for these three attributes in the create request, the Printer
3413 object MAY choose to change the value if the Printer object is able to compute a value which is more

3414 accurate than the client supplied value. The Printer object may be able to determine the correct value for
3415 these three attributes either right at job submission time or at any later point in time.

3416 4.3.18 job-impressions (integer(0:MAX))

3417 This attribute specifies the total size in number of impressions of the document(s) being submitted (see
3418 the definition of impression in section 13.2.5).

3419 As with "job-k-octets", this value MUST NOT include the multiplicative factors contributed by the
3420 number of copies specified by the "copies" attribute, independent of whether the device can process
3421 multiple copies without making multiple passes over the job or document data and independent of
3422 whether the output is collated or not. Thus the value is independent of the implementation and reflects
3423 the size of the document(s) measured in impressions independent of the number of copies.

3424 As with "job-k-octets", this value MUST also not include the multiplicative factor due to a copies
3425 instruction embedded in the document data. If the document data actually includes replications of the
3426 document data, this value will include such replication. In other words, this value is always the number
3427 of impressions in the source document data, rather than a measure of the number of impressions to be
3428 produced by the job.

3429 See the Note in the "job-k-octets" attribute that also applies to this attribute.

3430 4.3.19 job-media-sheets (integer(0:MAX))

3431 This attribute specifies the total number of media sheets to be produced for this job.

3432 Unlike the "job-k-octets" and the "job-impressions" attributes, this value MUST include the
3433 multiplicative factors contributed by the number of copies specified by the "copies" attribute and a
3434 'number of copies' instruction embedded in the document data, if any. This difference allows the system
3435 administrator to control the lower and upper bounds of both (1) the size of the document(s) with "job-k-
3436 octets-supported" and "job-impressions-supported" and (2) the size of the job with "job-media-sheets-
3437 supported".

3438 See the Note in the "job-k-octets" attribute that also applies to this attribute.

3439 4.3.20 job-k-octets-processed (integer(0:MAX))

3440 This attribute specifies the total number of octets processed in K octets, i.e., in units of 1024 octets so
3441 far. The value MUST be rounded up, so that a job between 1 and 1024 octets inclusive MUST be
3442 indicated as being 1, 1025 to 2048 inclusive MUST be 2, etc.

3443 For implementations where multiple copies are produced by the interpreter with only a single pass over
3444 the data, the final value MUST be equal to the value of the "job-k-octets" attribute. For implementations
3445 where multiple copies are produced by the interpreter by processing the data for each copy, the final
3446 value MUST be a multiple of the value of the "job-k-octets" attribute.

3447 Note: This attribute and the following two attributes ("job-impressions-completed" and "job-sheets-
3448 completed") are intended to be counters. That is, the value for a job that has not started processing
3449 MUST be 0. When the job's "job-state" is 'processing' or 'processing-stopped', this value is intended to
3450 contain the amount of the job that has been processed to the time at which the attributes are requested.

3451 4.3.21 job-impressions-completed (integer(0:MAX))

3452 This job attribute specifies the number of impressions completed for the job so far. For printing devices,
3453 the impressions completed includes interpreting, marking, and stacking the output.

3454 See the note in "job-k-octets-processed" which also applies to this attribute.

3455 4.3.22 job-media-sheets-completed (integer(0:MAX))

3456 This job attribute specifies the media-sheets completed marking and stacking for the entire job so far
3457 whether those sheets have been processed on one side or on both.

3458 See the note in "job-k-octets-processed" which also applies to this attribute.

3459 4.3.23 attributes-charset (charset)

3460 This REQUIRED attribute is populated using the value in the client supplied "attributes-charset"
3461 attribute in the create request. It identifies the charset (coded character set and encoding method) used
3462 by any Job attributes with attribute syntax 'text' and 'name' that were supplied by the client in the create
3463 request. See Section 3.1.4 for a complete description of the "attributes-charset" operation attribute.

3464 This attribute does not indicate the charset in which the 'text' and 'name' values are stored internally in
3465 the Job object. The internal charset is implementation-defined. The IPP object MUST convert from
3466 whatever the internal charset is to that being requested in an operation as specified in Section 3.1.4.

3467 4.3.24 attributes-natural-language (naturalLanguage)

3468 This REQUIRED attribute is populated using the value in the client supplied "attributes-natural-
3469 language" attribute in the create request. It identifies the natural language used for any Job attributes
3470 with attribute syntax 'text' and 'name' that were supplied by the client in the create request. See Section
3471 3.1.4 for a complete description of the "attributes-natural-language" operation attribute. See Sections
3472 4.1.1.2 and 4.1.2.2 for how a Natural Language Override may be supplied explicitly for each 'text' and
3473 'name' attribute value that differs from the value identified by the "attributes-natural-language" attribute.

3474 4.4 Printer Description Attributes

3475 These attributes form the attribute group called "printer-description". The following table summarizes
3476 these attributes, their syntax, and whether or not they are REQUIRED for a Printer object to support. If
3477 they are not indicated as REQUIRED, they are OPTIONAL. The maximum size in octets for 'text' and
3478 'name' attributes is indicated in parentheses.

3479 Note: How these attributes are set by an Administrator is outside the scope of this IPP/1.1 document.

	Attribute	Syntax	REQUIRED?
3480			
3481	Attribute	Syntax	REQUIRED?
3482			
3483	printer-uri-supported	1setOf uri	REQUIRED
3484			
3485	uri-security-supported	1setOf type2 keyword	REQUIRED
3486			
3487	printer-name	name (127)	REQUIRED
3488			
3489	printer-location	text (127)	
3490			
3491	printer-info	text (127)	
3492			
3493	printer-more-info	uri	
3494			
3495	printer-driver-installer	uri	
3496			
3497	printer-make-and-model	text (127)	
3498			
3499	printer-more-info-	uri	
3500	manufacturer		
3501			
3502	printer-state	type1 enum	REQUIRED
3503			
3504	printer-state-reasons	1setOf type2 keyword	
3505			
3506	printer-state-message	text (MAX)	
3507			
3508	operations-supported	1setOf type2 enum	REQUIRED
3509			
3510	charset-configured	charset	REQUIRED
3511			
3512	charset-supported	1setOf charset	REQUIRED
3513			
3514	natural-language-configured	naturalLanguage	REQUIRED
3515			
3516	generated-natural-language-	1setOf naturalLanguage	REQUIRED
3517	supported		
3518			
3519	document-format-default	mimeMediaType	REQUIRED
3520			
3521	document-format-supported	1setOf mimeMediaType	REQUIRED
3522			
3523	printer-is-accepting-jobs	boolean	REQUIRED
3524			
3525	queued-job-count	integer (0:MAX)	<u>REQUIRED</u>
3526	<u>RECOMMENDED</u>		
3527			
3528	printer-message-from-	text (127)	
3529	operator		

3530	+-----+-----+-----+
3531	color-supported boolean
3532	+-----+-----+-----+
3533	reference-uri-schemes- 1setOf uriScheme
3534	supported
3535	+-----+-----+-----+
3536	pdl-override-supported type2 keyword REQUIRED
3537	+-----+-----+-----+
3538	printer-up-time integer (1:MAX) REQUIRED
3539	+-----+-----+-----+
3540	printer-current-time dateTime
3541	+-----+-----+-----+
3542	multiple-operation-time-out integer (1:MAX)
3543	+-----+-----+-----+
3544	compression-supported 1setOf type3 keyword <u>REQUIRED</u>
3545	
3546	+-----+-----+-----+
3547	job-k-octets-supported rangeOfInteger (0:MAX)
3548	+-----+-----+-----+
3549	job-impressions-supported rangeOfInteger (0:MAX)
3550	+-----+-----+-----+
3551	job-media-sheets-supported rangeOfInteger (0:MAX)
3552	+-----+-----+-----+
3553	pages-per-minute integer(0:MAX)
3554	+-----+-----+-----+
3555	pages-per-minute-color integer(0:MAX)
3556	+-----+-----+-----+
3557	

3558 4.4.1 printer-uri-supported (1setOf uri)

3559 This REQUIRED Printer attribute contains at least one URI for the Printer object. It OPTIONALLY
 3560 contains more than one URI for the Printer object. An administrator determines a Printer object's
 3561 URI(s) and configures this attribute to contain those URIs by some means outside the scope of this
 3562 IPP/1.1 document. The precise format of this URI is implementation dependent and depends on the
 3563 protocol. See the next section for a description "uri-security-supported" which is the REQUIRED
 3564 companion attribute to this "printer-uri-supported" attribute. See section 2.4 on Printer object identity
 3565 and section 8.2 on security and URIs for more information.

3566 4.4.2 uri-security-supported (1setOf type2 keyword)

3567 This REQUIRED Printer attribute MUST have the same cardinality (contain the same number of values)
 3568 as the "printer-uri-supported" attribute. This attribute identifies the security mechanisms used for each
 3569 URI listed in the "printer-uri-supported" attribute. The "i th" value in "uri-security-supported"
 3570 corresponds to the "i th" value in "printer-uri-supported" and it describes the security mechanisms used
 3571 for accessing the Printer object via that URI. The following standard values are defined:

3572 'none': There are no secure communication channel protocols in use for the given URI.

3573 'ssl3': SSL3 [SSL] is the secure communications channel protocol in use for the given URI. For use
3574 in IPP/1.0.

3575 'tls': TLS [RFC2246] is the secure communications channel protocol in use for the given URI. For
3576 use in IPP/1.1.

3577

3578 This attribute is orthogonal to the specification of a client authentication mechanism. Specifically, 'none'
3579 does not exclude client authentication. Issue 21.

3580 Consider the following example. For a single Printer object, an administrator configures the "printer-uri-
3581 supported" and "uri-security-supported" attributes as follows:

3582 "printer-uri-supported": 'xxx://acme.com/open-use-printer', 'xxx://acme.com/restricted-use-printer',
3583 'xxx://acme.com/private-printer'

3584 "uri-security-supported": 'none', 'none', 'tls'

3585

3586 Note: 'xxx' is not a valid scheme. See the IPP/1.1 "Transport and Encoding" specification [ipp-pro] for
3587 the actual URI schemes to be used in object target attributes.

3588 In this case, one Printer object has three URIs.

- 3589 - For the first URI, 'xxx://acme.com/open-use-printer', the value 'none' in "uri-security-supported"
3590 indicates that there is no secure channel protocol configured to run under HTTP. The name
3591 implies that there is no Basic or Digest authentication being used, but it is up to the client to
3592 determine that while using HTTP underneath the IPP application protocol.
- 3593 - For the second URI, 'xxx://acme.com/restricted-use-printer', the value 'none' in "uri-security-
3594 supported" indicates that there is no secure channel protocol configured to run under HTTP. In
3595 this case, although the name does imply that there is some sort of Basic or Digest authentication
3596 being used within HTTP, it is up to the client to determine that while using HTTP and by
3597 processing any '401 Unauthorized' HTTP error messages.
- 3598 - For the third URI, 'xxx://acme.com/private-printer', the value 'tls' in "uri-security-supported"
3599 indicates that TLS is being used to secure the channel. The client SHOULD be prepared to use
3600 TLS framing to negotiate an acceptable ciphersuite to use while communicating with the Printer
3601 object. In this case, the name implies the use of a secure communications channel, but the fact is
3602 made explicit by the presence of the 'tls' value in "uri-security-supported". The client does not
3603 need to resort to understanding which security it must use by following naming conventions or by
3604 parsing the URI to determine which security mechanisms are implied.

3605

3606 It is expected that many IPP Printer objects will be configured to support only one channel (either
3607 configured to use TLS access or not), and will therefore only ever have one URI listed in the "printer-uri-
3608 supported" attribute. No matter the configuration of the Printer object (whether it has only one URI or
3609 more than one URI), a client MUST supply only one URI in the target "printer-uri" operation attribute.

3610 4.4.3 printer-name (name(127))

3611 This REQUIRED Printer attribute contains the name of the Printer object. It is a name that is more end-
3612 user friendly than a URI. An administrator determines a printer's name and sets this attribute to that
3613 name. This name may be the last part of the printer's URI or it may be unrelated. In non-US-English
3614 locales, a name may contain characters that are not allowed in a URI.

3615 4.4.4 printer-location (text(127))

3616 This Printer attribute identifies the location of the device. This could include things like: "in Room
3617 123A, second floor of building XYZ".

3618 4.4.5 printer-info (text(127))

3619 This Printer attribute identifies the descriptive information about this Printer object. This could include
3620 things like: "This printer can be used for printing color transparencies for HR presentations", or "Out of
3621 courtesy for others, please print only small (1-5 page) jobs at this printer", or even "This printer is going
3622 away on July 1, 1997, please find a new printer".

3623 4.4.6 printer-more-info (uri)

3624 This Printer attribute contains a URI used to obtain more information about this specific Printer object.
3625 For example, this could be an HTTP type URI referencing an HTML page accessible to a Web Browser.
3626 The information obtained from this URI is intended for end user consumption. Features outside the
3627 scope of IPP can be accessed from this URI. The information is intended to be specific to this printer
3628 instance and site specific services (e.g. job pricing, services offered, end user assistance). The device
3629 manufacturer may initially populate this attribute.

3630 4.4.7 printer-driver-installer (uri)

3631 This Printer attribute contains a URI to use to locate the driver installer for this Printer object. This
3632 attribute is intended for consumption by automata. The mechanics of print driver installation is outside
3633 the scope of this IPP/1.1 document. The device manufacturer may initially populate this attribute.

3634 4.4.8 printer-make-and-model (text(127))

3635 This Printer attribute identifies the make and model of the device. The device manufacturer may
3636 initially populate this attribute.

3637 4.4.9 printer-more-info-manufacturer (uri)

3638 This Printer attribute contains a URI used to obtain more information about this type of device. The
3639 information obtained from this URI is intended for end user consumption. Features outside the scope of
3640 IPP can be accessed from this URI (e.g., latest firmware, upgrades, print drivers, optional features
3641 available, details on color support). The information is intended to be germane to this printer without

3642 regard to site specific modifications or services. The device manufacturer may initially populate this
3643 attribute.

3644 4.4.10 printer-state (type1 enum)

3645 This REQUIRED Printer attribute identifies the current state of the device. The "printer-state reasons"
3646 attribute augments the "printer-state" attribute to give more detailed information about the Printer in the
3647 given printer state.

3648 A Printer object need only update this attribute before responding to an operation which requests the
3649 attribute; the Printer object NEED NOT update this attribute continually, since asynchronous event
3650 notification is not part of IPP/1.1. A Printer NEED NOT implement all values if they are not applicable
3651 to a given implementation.

3652 The following standard enum values are defined:

3653 Value	Symbolic Name and Description
------------	-------------------------------

3654

3655 '3'	'idle': If a Printer receives a job (whose required resources are ready) while in this state, 3656 such a job MUST transit into the 'processing' state immediately. If the "printer- 3657 state-reasons" attribute contains any reasons, they MUST be reasons that would 3658 not prevent a job from transiting into the 'processing' state immediately, e.g., 3659 'toner-low'. Note: if a Printer controls more than one output device, the above 3660 definition implies that a Printer is 'idle' if at least one output device is idle.
----------	--

3661

3662 '4'	'processing': If a Printer receives a job (whose required resources are ready) while in this 3663 state, such a job MUST transit into the 'pending' state immediately. Such a job 3664 MUST transit into the 'processing' state only after jobs ahead of it complete. If the 3665 "printer-state-reasons" attribute contains any reasons, they MUST be reasons that 3666 do not prevent the current job from printing, e.g. 'toner-low'. Note: if a Printer 3667 controls more than one output device, the above definition implies that a Printer is 3668 'processing' if at least one output device is processing, and none is idle.
----------	--

3669

3670 '5'	'stopped': If a Printer receives a job (whose required resources are ready) while in this 3671 state, such a job MUST transit into the 'pending' state immediately. Such a job 3672 MUST transit into the 'processing' state only after some human fixes the problem 3673 that stopped the printer and after jobs ahead of it complete processing. If 3674 supported, the "printer-state-reasons" attribute MUST contain at least one reason, 3675 e.g. 'media-jam', which prevents it from either processing the current job or 3676 transitioning a 'pending' job to the 'processing' state.
----------	--

3677

3678 Note: if a Printer controls more than one output device, the above definition
3679 implies that a Printer is 'stopped' only if all output devices are stopped. Also, it is
3680 tempting to define 'stopped' as when a sufficient number of output devices are
3681 stopped and leave it to an implementation to define the sufficient number. But
3682 such a rule complicates the definition of 'stopped' and 'processing'. For example,

3683 with this alternate definition of 'stopped', a job can move from 'pending' to
3684 'processing' without human intervention, even though the Printer is stopped.
3685

3686 4.4.11 printer-state-reasons (1setOf type2 keyword)

3687 This Printer attribute supplies additional detail about the device's state.

3688 Each keyword value MAY have a suffix to indicate its level of severity. The three levels are: report
3689 (least severe), warning, and error (most severe).

- 3690 - 'report': This suffix indicates that the reason is a "report". An implementation may choose to omit
3691 some or all reports. Some reports specify finer granularity about the printer state; others serve as
3692 a precursor to a warning. A report MUST contain nothing that could affect the printed output.
- 3693 - 'warning': This suffix indicates that the reason is a "warning". An implementation may choose to
3694 omit some or all warnings. Warnings serve as a precursor to an error. A warning MUST contain
3695 nothing that prevents a job from completing, though in some cases the output may be of lower
3696 quality.
- 3697 - 'error': This suffix indicates that the reason is an "error". An implementation MUST include all
3698 errors. If this attribute contains one or more errors, printer MUST be in the stopped state.
3699

3700 If the implementation does not add any one of the three suffixes, all parties MUST assume that the
3701 reason is an "error".

3702 If a Printer object controls more than one output device, each value of this attribute MAY apply to one or
3703 more of the output devices. An error on one output device that does not stop the Printer object as a
3704 whole MAY appear as a warning in the Printer's "printer-state-reasons attribute". If the "printer-state"
3705 for such a Printer has a value of 'stopped', then there MUST be an error reason among the values in the
3706 "printer-state-reasons" attribute.

3707 The following standard keyword values are defined:

- 3708 'other': The device has detected an error other than one listed in this document.
- 3709 'none': There are not reasons. This state reason is semantically equivalent to "printer-state-reasons"
3710 without any value.
- 3711 'media-needed': A tray has run out of media.
- 3712 'media-jam': The device has a media jam.
- 3713 'moving-to-paused': Someone has paused the Printer object using the Pause-Printer operation (see
3714 section 3.2.7) or other means, but the device(s) are taking an appreciable time to stop. Later,
3715 when all output has stopped, the "printer-state" becomes 'stopped', and the 'paused' value replaces
3716 the 'moving-to-paused' value in the "printer-state-reasons" attribute.
- 3717 'paused': Someone has paused the Printer object using the Pause-Printer operation (see section 3.2.7)
3718 or other means and the Printer object's "printer-state" is 'stopped'. In this state, a Printer MUST
3719 NOT produce printed output, but it MUST perform other operations requested by a client. If a
3720 Printer had been printing a job when the Printer was paused, the Printer MUST resume printing
3721 that job when the Printer is no longer paused and leave no evidence in the printed output of such
3722 a pause.

3723 'shutdown': Someone has removed a Printer object from service, and the device may be powered
3724 down or physically removed. In this state, a Printer object MUST NOT produce printed output,
3725 and unless the Printer object is realized by a print server that is still active, the Printer object
3726 MUST perform no other operations requested by a client, including returning this value. If a
3727 Printer object had been printing a job when it was shutdown, the Printer NEED NOT resume
3728 printing that job when the Printer is no longer shutdown. If the Printer resumes printing such a
3729 job, it may leave evidence in the printed output of such a shutdown, e.g. the part printed before
3730 the shutdown may be printed a second time after the shutdown.

3731 'connecting-to-device': The Printer object has scheduled a job on the output device and is in the
3732 process of connecting to a shared network output device (and might not be able to actually start
3733 printing the job for an arbitrarily long time depending on the usage of the output device by other
3734 servers on the network).

3735 'timed-out': The server was able to connect to the output device (or is always connected), but was
3736 unable to get a response from the output device.

3737 'stopping': The Printer object is in the process of stopping the device and will be stopped in a while.
3738 When the device is stopped, the Printer object will change the Printer object's state to 'stopped'.
3739 The 'stopping-warning' reason is never an error, even for a Printer with a single output device.
3740 When an output-device ceases accepting jobs, the Printer will have this reason while the output
3741 device completes printing.

3742 'stopped-partly': When a Printer object controls more than one output device, this reason indicates
3743 that one or more output devices are stopped. If the reason is a report, fewer than half of the
3744 output devices are stopped. If the reason is a warning, fewer than all of the output devices are
3745 stopped.

3746 'toner-low': The device is low on toner.

3747 'toner-empty': The device is out of toner.

3748 'spool-area-full': The limit of persistent storage allocated for spooling has been reached.

3749 'cover-open': One or more covers on the device are open.

3750 'interlock-open': One or more interlock devices on the printer are unlocked.

3751 'door-open': One or more doors on the device are open.

3752 'input-tray-missing': One or more input trays are not in the device.

3753 'media-low': At least one input tray is low on media.

3754 'media-empty': At least one input tray is empty.

3755 'output-tray-missing': One or more output trays are not in the device

3756 'output-area-almost-full': One or more output area is almost full (e.g. tray, stacker, collator).

3757 'output-area-full': One or more output area is full. (e.g. tray, stacker, collator)

3758 'marker-supply-low': The device is low on at least one marker supply. (e.g. toner, ink, ribbon)

3759 'marker-supply-empty': The device is out of at least one marker supply. (e.g. toner, ink, ribbon)

3760 'marker-waste-almost-full': The device marker supply waste receptacle is almost full.

3761 'marker-waste-full': The device marker supply waste receptacle is full.

3762 'fuser-over-temp': The fuser temperature is above normal.

3763 'fuser-under-temp': The fuser temperature is below normal.

3764 'opc-near-eol': The optical photo conductor is near end of life.

3765 'opc-life-over': The optical photo conductor is no longer functioning.

3766 'developer-low': The device is low on developer.

3767 'developer-empty': The device is out of developer.

3768 "interpreter-resource-unavailable": An interpreter resource is unavailable (i.e. font, form)
3769

3770 4.4.12 printer-state-message (text(MAX))

3771 This Printer attribute specifies the additional information about the printer state and printer state reasons
3772 in human readable text. If the Printer object supports this attribute, the Printer object MUST be able to
3773 generate this message in any of the natural languages identified by the Printer's "generated-natural-
3774 language-supported" attribute (see the "attributes-natural-language" operation attribute specified in
3775 Section 3.1.4.1).

3776 4.4.13 operations-supported (1setOf type2 enum)

3777 This REQUIRED Printer attribute specifies the set of supported operations for this Printer object and
3778 contained Job objects.

3779 Note: This attribute is encoded as any other enum attribute syntax according to [IPP-PRO] as 32-bits.
3780 However, all 32-bit enum values for this attribute MUST NOT exceed 0x00008FFF, since these same
3781 values are also passed in two octets in the "operation-id" parameter (see section 3.1.1) in each Protocol
3782 request with the two high order octets omitted in order to indicate the operation being performed [IPP-
3783 PRO].

3784 The following standard enum and "operation-id" (see section 3.1.2) values are defined:

3785	Value	Operation Name
3786	-----	-----
3787		
3788	0x0000	reserved, not used
3789	0x0001	reserved, not used
3790	0x0002	Print-Job
3791	0x0003	Print-URI
3792	0x0004	Validate-Job
3793	0x0005	Create-Job
3794	0x0006	Send-Document
3795	0x0007	Send-URI
3796	0x0008	Cancel-Job
3797	0x0009	Get-Job-Attributes
3798	0x000A	Get-Jobs
3799	0x000B	Get-Printer-Attributes
3800	0x000C	Hold-Job
3801	0x000D	Release-Job
3802	0x000E	Restart-Job
3803	0x000F	reserved for a future operation
3804	0x0010	Pause-Printer
3805	0x0011	Resume-Printer
3806	0x0012	Purge-Jobs

3807 0x00013-0x3FFF reserved for future operations
3808 0x4000-0x8FFF reserved for private extensions
3809

3810 This allows for certain vendors to implement private extensions that are guaranteed to not conflict with
3811 future registered extensions. However, there is no guarantee that two or more private extensions will not
3812 conflict.

3813 4.4.14 charset-configured (charset)

3814 This REQUIRED Printer attribute identifies the charset that the Printer object has been configured to
3815 represent 'text' and 'name' Printer attributes that are set by the operator, system administrator, or
3816 manufacturer, i.e., for "printer-name" (name), "printer-location" (text), "printer-info" (text), and "printer-
3817 make-and-model" (text). Therefore, the value of the Printer object's "charset-configured" attribute
3818 MUST also be among the values of the Printer object's "charset-supported" attribute.

3819 4.4.15 charset-supported (1setOf charset)

3820 This REQUIRED Printer attribute identifies the set of charsets that the Printer and contained Job objects
3821 support in attributes with attribute syntax 'text' and 'name'. At least the value 'utf-8' MUST be present,
3822 since IPP objects MUST support the UTF-8 [RFC2279] charset. If a Printer object supports a charset, it
3823 means that for all attributes of syntax 'text' and 'name' the IPP object MUST (1) accept the charset in
3824 requests and return the charset in responses as needed.

3825 If more charsets than UTF-8 are supported, the IPP object MUST perform charset conversion between
3826 the charsets as described in Section 3.2.1.2.

3827 4.4.16 natural-language-configured (naturalLanguage)

3828 This REQUIRED Printer attribute identifies the natural language that the Printer object has been
3829 configured to represent 'text' and 'name' Printer attributes that are set by the operator, system
3830 administrator, or manufacturer, i.e., for "printer-name" (name), "printer-location" (text), "printer-info"
3831 (text), and "printer-make-and-model" (text). When returning these Printer attributes, the Printer object
3832 MAY return them in the configured natural language specified by this attribute, instead of the natural
3833 language requested by the client in the "attributes-natural-language" operation attribute. See Section
3834 3.1.4.1 for the specification of the OPTIONAL multiple natural language support. Therefore, the value
3835 of the Printer object's "natural-language-configured" attribute MUST also be among the values of the
3836 Printer object's "generated-natural-language-supported" attribute.

3837 4.4.17 generated-natural-language-supported (1setOf naturalLanguage)

3838 This REQUIRED Printer attribute identifies the natural language(s) that the Printer object and contained
3839 Job objects support in attributes with attribute syntax 'text' and 'name'. The natural language(s)
3840 supported depends on implementation and/or configuration. Unlike charsets, IPP objects MUST accept
3841 requests with any natural language or any Natural Language Override whether the natural language is
3842 supported or not.

3843 If a Printer object supports a natural language, it means that for any of the attributes for which the Printer
3844 or Job object generates messages, i.e., for the "job-state-message" and "printer-state-message" attributes
3845 and Operation Messages (see Section 3.1.5) in operation responses, the Printer and Job objects MUST be
3846 able to generate messages in any of the Printer's supported natural languages. See section 3.1.4 for the
3847 specification of 'text' and 'name' attributes in operation requests and responses.

3848 Note: A Printer object that supports multiple natural languages, often has separate catalogs of messages,
3849 one for each natural language supported.

3850 4.4.18 document-format-default (mimeMediaType)

3851 This REQUIRED Printer attribute identifies the document format that the Printer object has been
3852 configured to assume if the client does not supply a "document-format" operation attribute in any of the
3853 operation requests that supply document data. The standard values for this attribute are Internet Media
3854 types (sometimes called MIME types). For further details see the description of the 'mimeMediaType'
3855 attribute syntax in Section 4.1.9.

3856 4.4.19 document-format-supported (1setOf mimeMediaType)

3857 This REQUIRED Printer attribute identifies the set of document formats that the Printer object and
3858 contained Job objects can support. For further details see the description of the 'mimeMediaType'
3859 attribute syntax in Section 4.1.9.

3860 4.4.20 printer-is-accepting-jobs (boolean)

3861 This REQUIRED Printer attribute indicates whether the printer is currently able to accept jobs, i.e., is
3862 accepting Print-Job, Print-URI, and Create-Job requests. If the value is 'true', the printer is accepting
3863 jobs. If the value is 'false', the Printer object is currently rejecting any jobs submitted to it. In this case,
3864 the Printer object returns the 'server-error-not-accepting-jobs' status code.

3865 Note: This value is independent of the "printer-state" and "printer-state-reasons" attributes because its
3866 value does not affect the current job; rather it affects future jobs. This attribute may cause the Printer to
3867 reject jobs when the "printer-state" is 'idle' or it may cause the Printer object to accept jobs when the
3868 "printer-state" is 'stopped'.

3869 4.4.21 queued-job-count (integer(0:MAX))

3870 This ~~REQUIRED~~ ~~RECOMMENDED~~ Printer attribute contains a count of the number of jobs that are
3871 either 'pending', 'processing', 'pending-held', or 'processing-stopped' and is set by the Printer object. **Issue**
3872 **29**

3873 4.4.22 printer-message-from-operator (text(127))

3874 This Printer attribute provides a message from an operator, system administrator or "intelligent" process
3875 to indicate to the end user information or status of the printer, such as why it is unavailable or when it is
3876 expected to be available.

3877 4.4.23 color-supported (boolean)

3878 This Printer attribute identifies whether the device is capable of any type of color printing at all,
3879 including highlight color. All document instructions having to do with color are embedded within the
3880 document PDL (none are external IPP attributes in IPP/1.1).

3881 Note: end-users are able to determine the nature and details of the color support by querying the
3882 "printer-more-info-manufacturer" Printer attribute.

3883 4.4.24 reference-uri-schemes-supported (1setOf uriScheme)

3884 This Printer attribute specifies which URI schemes are supported for use in the "document-uri" operation
3885 attribute of the Print-URI or Send-URI operation. If a Printer object supports these optional operations,
3886 it MUST support the "reference-uri-schemes-supported" Printer attribute with at least the following
3887 schemed URI value:

3888 'ftp': The Printer object will use an FTP 'get' operation as defined in RFC 2228 [RFC2228] using
3889 FTP URLs as defined by [RFC2396] and[RFC2316].

3890
3891 The Printer object MAY OPTIONALLY support other URI schemes (see section 4.1.6).

3892 4.4.25 pdl-override-supported (type2 keyword)

3893 This REQUIRED Printer attribute expresses the ability for a particular Printer implementation to either
3894 attempt to override document data instructions with IPP attributes or not.

3895 This attribute takes on the following values:

- 3896 - 'attempted': This value indicates that the Printer object attempts to make the IPP attribute values
3897 take precedence over embedded instructions in the document data, however there is no guarantee.
- 3898 - 'not-attempted': This value indicates that the Printer object makes no attempt to make the IPP
3899 attribute values take precedence over embedded instructions in the document data.

3900

3901 Section 16 contains a full description of how this attribute interacts with and affects other IPP attributes,
3902 especially the "ipp-attribute-fidelity" attribute.

3903 4.4.26 printer-up-time (integer(1:MAX))

3904 This REQUIRED Printer attribute indicates the amount of time (in seconds) that this instance of this
3905 Printer implementation has been up and running. This value is used to populate the Job attributes "time-

3906 at-creation", "time-at-processing", and "time-at-completed". These time values are all measured in
3907 seconds and all have meaning only relative to this attribute, "printer-up-time". The value is a
3908 monotonically increasing value starting from 1 when the Printer object is started-up (initialized, booted,
3909 etc.).

3910 If the Printer object goes down at some value 'n', and comes back up, the implementation MAY:

- 3911 1. Know how long it has been down, and resume at some value greater than 'n', or
- 3912 2. Restart from 1.

3913

3914 In the first case, the Printer SHOULD not tweak any existing related Job attributes ("time-at-creation",
3915 "time-at-processing", and "time-at-completed"). In the second case, the Printer object SHOULD reset
3916 those attributes to 0. If a client queries a time-related Job attribute and finds the value to be 0, the client
3917 MUST assume that the Job was submitted in some life other than the Printer's current life.

3918 4.4.27 printer-current-time (dateTime)

3919 This Printer attribute indicates the current absolute wall-clock time. If an implementation supports this
3920 attribute, then a client could calculate the absolute wall-clock time each Job's "time-at-creation", "time-
3921 at-processing", and "time-at-completed" attributes by using both "printer-up-time" and this attribute,
3922 "printer-current-time". If an implementation does not support this attribute, a client can only calculate
3923 the relative time of certain events based on the REQUIRED "printer-up-time" attribute.

3924 4.4.28 multiple-operation-time-out (integer(1:MAX))

3925 This Printer attribute identifies the minimum time (in seconds) that the Printer object waits for
3926 additional Send-Document or Send-URI operations to follow a still-open multi-document Job object
3927 before taking any recovery actions, such as the ones indicated in section 3.3.1. If the Printer object
3928 supports the Create-Job operation (see section 3.2.4), it MUST support this attribute.

3929 It is RECOMMENDED that vendors supply a value for this attribute that is between 60 and 240
3930 seconds. An implementation MAY allow a system administrator to set this attribute (by means outside
3931 this IPP/1.1 document). If so, the system administrator MAY be able to set values outside this range.

3932 4.4.29 compression-supported (1setOf type3 keyword)

3933 This **REQUIRED** Printer attribute identifies the set of supported compression algorithms for document
3934 data. Compression only applies to the document data; compression does not apply to the encoding of the
3935 IPP operation itself. The supported values are used to validate the client supplied "compression"
3936 operation attributes in Print-Job, Send-Document, and Send-URI requests. **Issue 28**

3937 Standard values are :

- 3938 'none': no compression is used.
- 3939 'deflate': ZIP public domain inflate/deflate) compression technology
- 3940 'gzip' GNU zip compression technology described in RFC 1952 [RFC1952].

3941 'compress': UNIX compression technology

3942

3943 4.4.30 job-k-octets-supported (rangeOfInteger(0:MAX))

3944 This Printer attribute specifies the upper and lower bounds of total sizes of jobs in K octets, i.e., in units
3945 of 1024 octets. The supported values are used to validate the client supplied "job-k-octets" operation
3946 attributes in create requests. The corresponding job description attribute "job-k-octets" is defined in
3947 section 4.3.17.

3948 4.4.31 job-impressions-supported (rangeOfInteger(0:MAX))

3949 This Printer attribute specifies the upper and lower bounds for the number of impressions per job. The
3950 supported values are used to validate the client supplied "job-impressions" operation attributes in create
3951 requests. The corresponding job description attribute "job-impressions" is defined in section 4.3.18.

3952 4.4.32 job-media-sheets-supported (rangeOfInteger(0:MAX))

3953 This Printer attribute specifies the upper and lower bounds for the number of media sheets per job. The
3954 supported values are used to validate the client supplied "job-media-sheets" operation attributes in create
3955 requests. The corresponding Job attribute "job-media-sheets" is defined in section 4.3.19.

3956 4.4.33 pages-per-minute (integer(0:MAX))

3957 This Printer attributes specifies the nominal number of pages per minute to the nearest whole number
3958 which may be generated by this printer (e.g., simplex, black-and-white). This attribute is informative,
3959 not a service guarantee. Generally, it is the value used in the marketing literature to describe the device.

3960 A value of 0 indicates a device that takes more than two minutes to process a page.

3961 4.4.34 pages-per-minute-color (integer(0:MAX))

3962 This Printer attributes specifies the nominal number of pages per minute to the nearest whole number
3963 which may be generated by this printer when printing color (e.g., simplex, color). For purposes of this
3964 attribute, "color" means the same as for the "color-supported" attribute, namely, the device is capable of
3965 any type of color printing at all, including highlight color. This attribute is informative, not a service
3966 guarantee. Generally, it is the value used in the marketing literature to describe the color capabilities of
3967 this device.

3968 A value of 0 indicates a device that takes more than two minutes to process a page.

3969 Note: If a color device has several color modes, it MAY use the pages-per-minute value for this
3970 attribute that corresponds to the mode that produces the highest number.

3971 Black and white only printers MUST NOT support this attribute. If this attribute is present, then the
3972 "color-supported" Printer description attribute MUST be present and have a 'true' value.

3973 Note: The values of these two attributes returned by the Get-Printer-Attributes operation MAY be
3974 affected by the "document-format" attribute supplied by the client in the Get-Printer-Attributes request.
3975 In other words, the implementation MAY have different speeds depending on the document format
3976 being processed. See section 3.2.5.1 Get-Printer-Attributes.

3977 5. Conformance

3978 This section describes conformance issues and requirements. This document introduces model entities
3979 such as objects, operations, attributes, attribute syntaxes, and attribute values. These conformance
3980 sections describe the conformance requirements which apply to these model entities.

3981 5.1 Client Conformance Requirements

3982 A conforming client **MUST** support all **REQUIRED** operations as defined in this document. For each
3983 attribute included in an operation request, a conforming client **MUST** supply a value whose type and
3984 value syntax conforms to the requirements of the Model document as specified in Sections 3 and 3.3.5.
3985 A conforming client **MAY** supply any registered extensions and/or private extensions in an operation
3986 request, as long as they meet the requirements in Section 6.

3987 Otherwise, there are no conformance requirements placed on the user interfaces provided by IPP clients
3988 or their applications. For example, one application might not allow an end user to submit multiple
3989 documents per job, while another does. One application might first query a Printer object in order to
3990 supply a graphical user interface (GUI) dialogue box with supported and default values whereas a
3991 different implementation might not.

3992 When sending a request, an IPP client **NEED NOT** supply any attributes that are indicated as
3993 **OPTIONALLY** supplied by the client.

3994 A client **MUST** be able to accept any of the attribute syntaxes defined in Section 4.1, including their full
3995 range, that may be returned to it in a response from a Printer object. In particular for each attribute that
3996 the client supports whose attribute syntax is 'text', the client **MUST** accept and process both the
3997 'textWithoutLanguage' and 'textWithLanguage' forms. Similarly, for each attribute that the client
3998 supports whose attribute syntax is 'name', the client **MUST** accept and process both the
3999 'nameWithoutLanguage' and 'nameWithLanguage' forms. For presentation purposes, truncation of long
4000 attribute values is not recommended. A recommended approach would be for the client implementation
4001 to allow the user to scroll through long attribute values.

4002 A ~~query~~ response may contain attribute groups, attributes, and values that the client does not expect.
4003 Therefore, a client implementation **MUST** gracefully handle such responses and not refuse to inter-
4004 operate with a conforming Printer that is returning extended registered or private attributes and/or
4005 attribute values that conform to Section 6. Clients may choose to ignore any parameters, attributes, or
4006 values that they do not understand.

4007 A client MUST NOT allow a channel to close because of a transport layer time-out while sending data to
 4008 a printer (i.e. flow-controlled off) for whatever reason, e.g. 'out of paper' or 'job ahead hasn't freed up
 4009 enough memory'. However, the layer that launched the print submission (e.g. an end user) MAY close
 4010 the channel in order to cancel the job. When a client closes a channel, a printer MAY print all or part of
 4011 the received portion of the document. Issues #4 and 5

4012 5.2 IPP Object Conformance Requirements

4013 This section specifies the conformance requirements for conforming implementations with respect to
 4014 objects, operations, and attributes.

4015 5.2.1 Objects

4016 Conforming implementations MUST implement all of the model objects as defined in this specification
 4017 in the indicated sections:

4018 Section 2.1 - Printer Object

4019 Section 2.2 - Job Object

4020

4021 5.2.2 Operations

4022 Conforming IPP object implementations MUST implement all of the REQUIRED model operations,
 4023 including REQUIRED responses, as defined in this specification in the indicated sections:

4024 For a Printer object:

4025 Print-Job (section 3.2.1) REQUIRED

4026 Print-URI (section 3.2.2) OPTIONAL

4027 Validate-Job (section 3.2.3) REQUIRED

4028 Create-Job (section 3.2.4) OPTIONAL

4029 Get-Printer-Attributes (section 3.2.5) REQUIRED

4030 Get-Jobs (section 3.2.6) REQUIRED

4031 Pause-Printer (section 3.2.7) OPTIONAL

4032 Resume-Printer (section 3.2.8) OPTIONAL

4033 Purge-Jobs (section 3.2.9) OPTIONAL

4034

4035 For a Job object:

4036 Send-Document (section 3.3.1) OPTIONAL

4037 Send-URI (section 3.3.2) OPTIONAL

4038 Cancel-Job (section 3.3.3) REQUIRED

4039 Get-Job-Attributes (section 3.3.4) REQUIRED

4040 Hold-Job (section 3.3.5) OPTIONAL

4041 Release-Job (section 3.3.6) OPTIONAL

4042 Restart-Job (section 3.3.7) OPTIONAL

4043

4044 Conforming IPP objects MUST support all REQUIRED operation attributes and all values of such
4045 attributes if so indicated in the description. Conforming IPP objects MUST ignore all unsupported or
4046 unknown operation attributes or operation attribute groups received in a request, but MUST reject a
4047 request that contains a supported operation attribute that contains an unsupported value.

4048 Conforming IPP objects MAY send attributes groups or attributes within new or existing attribute
4049 groups that are extensions to this standard. **Issue 25 and 26**

4050 The following section on object attributes specifies the support required for object attributes.

4051 5.2.3 IPP Object Attributes

4052 Conforming IPP objects MUST support all of the REQUIRED object attributes, as defined in this
4053 specification in the indicated sections.

4054 If an object supports an attribute, it MUST support only those values specified in this document or
4055 through the extension mechanism described in section 5.2.4. It MAY support any non-empty subset of
4056 these values. That is, it MUST support at least one of the specified values and at most all of them.

4057 5.2.4 Versions

4058 Clients MUST support version 1.1 and MAY also support version 1.0. IPP objects MUST support both
4059 version 1.0 and 1.1. See section 3.1.7.

4060 5.2.5 Extensions

4061 A conforming IPP object MAY support registered extensions and private extensions, as long as they
4062 meet the requirements specified in Section 6.

4063 For each attribute included in an operation response, a conforming IPP object MUST return a value
4064 whose type and value syntax conforms to the requirement of the Model document as specified in
4065 Sections 3 and 3.3.5.

4066 5.2.6 Attribute Syntaxes

4067 An IPP object MUST be able to accept any of the attribute syntaxes defined in Section 4.1, including
4068 their full range, in any operation in which a client may supply attributes or the system administrator may
4069 configure attributes (by means outside the scope of this IPP/1.1 document). In particular for each
4070 attribute that the IPP object supports whose attribute syntax is 'text', the IPP object MUST accept and
4071 process both the 'textWithoutLanguage' and 'textWithLanguage' forms. Similarly, for each attribute that
4072 the IPP object supports whose attribute syntax is 'name', the IPP object MUST accept and process both
4073 the 'nameWithoutLanguage' and 'nameWithLanguage' forms. Furthermore, an IPP object MUST return
4074 attributes to the client in operation responses that conform to the syntax specified in Section 4.1,
4075 including their full range if supplied previously by a client.

4076 5.3 Charset and Natural Language Requirements

4077 All clients and IPP objects **MUST** support the 'utf-8' charset as defined in section 4.1.7.

4078 IPP objects **MUST** be able to accept any client request which correctly uses the "attributes-natural-
4079 language" operation attribute or the Natural Language Override mechanism on any individual attribute
4080 whether or not the natural language is supported by the IPP object. If an IPP object supports a natural
4081 language, then it **MUST** be able to translate (perhaps by table lookup) all generated 'text' or 'name'
4082 attribute values into one of the supported languages (see section 3.1.4). That is, the IPP object that
4083 supports a natural language **NEED NOT** be a general purpose translator of any arbitrary 'text' or 'name'
4084 value supplied by the client into that natural language. However, the object **MUST** be able to translate
4085 (automatically generate) any of its own attribute values and messages into that natural language.

4086 5.4 Security Conformance Requirements

4087 Conforming IPP Printer objects **SHOULD** support Transport Layer Security (TLS) protocol Version 1
4088 (TLS) [RFC2246] access, **MAY** support access without TLS, or **MAY** support both means of access.

4089 Conforming IPP clients **SHOULD** support TLS access and non-TLS access. Note: This client
4090 recommendation to support both means that conforming IPP clients will be able to inter-operate with any
4091 IPP Printer object.

4092 For a detailed discussion of security considerations and the IPP application security profile required for
4093 TLS support, see section 8.

4094 6. IANA Considerations (registered and private extensions)

4095 This section describes how IPP can be extended to allow the following registered and private extensions
4096 to IPP:

- 4097 1. keyword attribute values
- 4098 2. enum attribute values
- 4099 3. attributes
- 4100 4. attribute syntaxes
- 4101 5. operations
- 4102 6. attribute groups
- 4103 7. status codes

4104

4105 Extensions registered for use with IPP/1.1 are **OPTIONAL** for client and IPP object conformance to the
4106 IPP/1.1 Model specification.

4107 These extension procedures are aligned with the guidelines as set forth by the IESG [IANA-CON].
4108 Section 12 describes how to propose new registrations for consideration. IANA will reject registration
4109 proposals that leave out required information or do not follow the appropriate format described in

4110 Section 12. IPP/1.1 may also be extended by an appropriate RFC that specifies any of the above
4111 extensions.

4112 6.1 Typed 'keyword' and 'enum' Extensions

4113 IPP allows for 'keyword' and 'enum' extensions (see sections 4.1.2.3 and 4.1.4). This document uses
4114 prefixes to the 'keyword' and 'enum' basic attribute syntax type in order to communicate extra
4115 information to the reader through its name. This extra information is not represented in the protocol
4116 because it is unimportant to a client or Printer object. The list below describes the prefixes and their
4117 meaning.

4118 "type1": The IPP specification must be revised to add a new keyword or a new enum. No private
4119 keywords or enums are allowed.

4120
4121 "type2": Implementers can, at any time, add new keyword or enum values by proposing the
4122 complete specification to IANA:

4123
4124 iana@iana.org

4125
4126 IANA will forward the registration proposal to the IPP Designated Expert who will review the
4127 proposal with a mailing list that the Designated Expert keeps for this purpose. Initially, that list
4128 will be the mailing list used by the IPP WG:

4129
4130 ipp@pwg.org

4131
4132 even after the IPP WG is disbanded as permitted by [IANA-CON]. The IPP Designated Expert
4133 is appointed by the IESG Area Director responsible for IPP, according to [IANA-CON].

4134
4135 When a type2 keyword or enum is approved, the IPP Designated Expert becomes the point of
4136 contact for any future maintenance that might be required for that registration.

4137
4138 "type3": Implementers can, at any time, add new keyword and enum values by submitting the
4139 complete specification to IANA as for type2 who will forward the proposal to the IPP Designated
4140 Expert. While no additional technical review is required, the IPP Designated Expert may, at
4141 his/her discretion, forward the proposal to the same mailing list as for type2 registrations for
4142 advice and comment.

4143
4144 When a type3 keyword or enum is approved by the IPP Designated Expert, the original proposer
4145 becomes the point of contact for any future maintenance that might be required for that
4146 registration.

4147
4148 For type2 and type3 keywords, the proposer includes the name of the keyword in the registration
4149 proposal and the name is part of the technical review.

4150 After type2 and type3 enums specifications are approved, the IPP Designated Expert in consultation with
4151 IANA assigns the next available enum number for each enum value.

4152 IANA will publish approved type2 and type3 keyword and enum attributes value registration
4153 specifications in:

4154 ftp.isi.edu/iana/assignments/ipp/attribute-values/xxx/yyy.txt

4155 where xxx is the attribute name that specifies the initial values and yyy.txt is a descriptive file name that
4156 contains one or more enums or keywords approved at the same time. For example, if several additional
4157 enums for stapling are approved for use with the "finishings" attribute (and "finishings-default" and
4158 "finishings-supported" attributes), IANA will publish the additional values in the file:

4159 ftp.isi.edu/iana/assignments/ipp/attribute-values/finishings/stapling.txt.

4160 Note: Some attributes are defined to be: 'type3 keywords' | 'name' which allows for attribute values to be
4161 extended by a site administrator with administrator defined names. Such names are not registered with
4162 IANA.

4163 By definition, each of the three types above assert some sort of registry or review process in order for
4164 extensions to be considered valid. Each higher numbered level (1, 2, 3) tends to be decreasingly less
4165 stringent than the previous level. Therefore, any typeN value MAY be registered using a process for
4166 some typeM where M is less than N, however such registration is NOT REQUIRED. For example, a
4167 type3 value MAY be registered in a type 1 manner (by being included in a future version of an IPP
4168 specification), however, it is NOT REQUIRED.

4169 This specification defines keyword and enum values for all of the above types, including type3
4170 keywords.

4171 For private (unregistered) keyword extensions, implementers SHOULD use keywords with a suitable
4172 distinguishing prefix, such as "xxx-" where xxx is the (lowercase) fully qualified company name
4173 registered with IANA for use in domain names [RFC1035]. For example, if the company XYZ Corp.
4174 had obtained the domain name "XYZ.com", then a private keyword 'abc' would be: 'xyz.com-abc'.

4175 Note: RFC 1035 [RFC1035] indicates that while upper and lower case letters are allowed in domain
4176 names, no significance is attached to the case. That is, two names with the same spelling but different
4177 case are to be treated as if identical. Also, the labels in a domain name must follow the rules for
4178 ARPANET host names: They must start with a letter, end with a letter or digit, and have as interior
4179 characters only letters, digits, and hyphen. Labels must be 63 characters or less. Labels are separated by
4180 the "." character.

4181 For private (unregistered) enum extension, implementers MUST use values in the reserved integer range
4182 which is 2^{30} to $2^{31}-1$.

4183 6.2 Attribute Extensibility

4184 Attribute names are type2 keywords. Therefore, new attributes may be registered and have the same
4185 status as attributes in this document by following the type2 extension rules. For private (unregistered)
4186 attribute extensions, implementers SHOULD use keywords with a suitable distinguishing prefix as
4187 described in Section 6.1.

4188 IANA will publish approved attribute registration specifications as separate files:

4189 `ftp.isi.edu/iana/assignments/ipp/attributes/xxx-yyy.txt`

4190 where "xxx-yyy" is the new attribute name.

4191 If a new Printer object attribute is defined and its values can be affected by a specific document format,
4192 its specification needs to contain the following sentence:

4193 "The value of this attribute returned in a Get-Printer-Attributes response MAY depend on the
4194 "document-format" attribute supplied (see Section 3.2.5.1)."

4195 If the specification does not, then its value in the Get-Printer-Attributes response MUST NOT depend on
4196 the "document-format" supplied in the request. When a new Job Template attribute is registered, the
4197 value of the Printer attributes MAY vary with "document-format" supplied in the request without the
4198 specification having to indicate so.

4199 6.3 Attribute Syntax Extensibility

4200 Attribute syntaxes are like type2 enums. Therefore, new attribute syntaxes may be registered and have
4201 the same status as attribute syntaxes in this document by following the type2 extension rules described in
4202 Section 6.1. The value codes that identify each of the attribute syntaxes are assigned in the "Encoding
4203 and Transport" specification [IPP-PRO], including a designated range for private, experimental use.

4204 For attribute syntaxes, the IPP Designated Expert in consultation with IANA assigns the next attribute
4205 syntax code in the appropriate range as specified in [IPP-PRO]. IANA will publish approved attribute
4206 syntax registration specifications as separate files:

4207 `ftp.isi.edu/iana/assignments/ipp/attribute-syntaxes/xxx-yyy.txt`

4208 where 'xxx-yyy' is the new attribute syntax name.

4209 6.4 Operation Extensibility

4210 Operations may also be registered following the type2 procedures described in Section 6.1, though major
4211 new operations will usually be done by a new standards track RFC that augments this document. For
4212 private (unregistered) operation extensions, implementers MUST use the range for the "operation-id" in
4213 requests specified in Section 4.4.13 "operations-supported" Printer attribute.

4214 For operations, the IPP Designated Expert in consultation with IANA assigns the next operation-id code
4215 as specified in Section 4.4.13. IANA will publish approved operation registration specifications as
4216 separate files:

4217 `ftp.isi.edu/iana/assignments/ipp/operations/Xxx-Yyy.txt`

4218 where "Xxx-Yyy" is the new operation name.

4219 6.5 Attribute Groups

4220 Attribute groups passed in requests and responses may be registered following the type2 procedures
4221 described in Section 6.1. The tags that identify each of the attribute groups are assigned in [IPP-PRO].

4222 For attribute groups, the IPP Designated Expert in consultation with IANA assigns the next attribute
4223 group tag code in the appropriate range as specified in [IPP-PRO]. IANA will publish approved
4224 attribute group registration specifications as separate files:

4225 `ftp.isi.edu/iana/assignments/ipp/attribute-group-tags/xxx-yyy-tag.txt`

4226 where 'xxx-yyy-tag' is the new attribute group tag name.

4227 6.6 Status Code Extensibility

4228 Operation status codes may also be registered following the type2 procedures described in Section 6.1.
4229 The values for status codes are allocated in ranges as specified in Section 14 for each status code class:

4230 "informational" - Request received, continuing process

4231 "successful" - The action was successfully received, understood, and accepted

4232 "redirection" - Further action must be taken in order to complete the request

4233 "client-error" - The request contains bad syntax or cannot be fulfilled

4234 "server-error" - The IPP object failed to fulfill an apparently valid request

4235

4236 For private (unregistered) operation status code extensions, implementers MUST use the top of each
4237 range as specified in Section 14.

4238 For operation status codes, the IPP Designated Expert in consultation with IANA assigns the next status
4239 code in the appropriate class range as specified in Section 14. IANA will publish approved status code
4240 registration specifications as separate files:

4241 `ftp.isi.edu/iana/assignments/ipp/status-codes/xxx-yyy.txt`

4242 where "xxx-yyy" is the new operation status code keyword.

4243 6.7 Registration of MIME types/sub-types for document-formats

4244 The "document-format" attribute's syntax is 'mimeMediaType'. This means that valid values are Internet
4245 Media Types (see Section 4.1.9). RFC 2045 [RFC2045] defines the syntax for valid Internet media
4246 types. IANA is the registry for all Internet media types.

4247 6.8 Registration of charsets for use in 'charset' attribute values

4248 The "attributes-charset" attribute's syntax is 'charset'. This means that valid values are charsets names.
4249 When a charset in the IANA registry has more than one name (alias), the name labeled as "(preferred
4250 MIME name)", if present, MUST be used (see Section 4.1.7). IANA is the registry for charsets
4251 following the procedures of [RFC2278].

4252 7. Internationalization Considerations

4253 Some of the attributes have values that are text strings and names which are intended for human
4254 understanding rather than machine understanding (see the 'text' and 'name' attribute syntaxes in Sections
4255 4.1.1 and 4.1.2).

4256 In each operation request, the client

- 4257 - identifies the charset and natural language of the request which affects each supplied 'text' and
- 4258 'name' attribute value, and
- 4259 - requests the charset and natural language for attributes returned by the IPP object in operation
- 4260 responses (as described in Section 3.1.4.1).

4261

4262 In addition, the client MAY separately and individually identify the Natural Language Override of a
4263 supplied 'text' or 'name' attribute using the 'textWithLanguage' and 'nameWithLanguage' technique
4264 described section 4.1.1.2 and 4.1.2.2 respectively.

4265 All IPP objects MUST support the UTF-8 [RFC2279] charset in all 'text' and 'name' attributes supported.
4266 If an IPP object supports more than the UTF-8 charset, the object MUST convert between them in order
4267 to return the requested charset to the client according to Section 3.1.4.2. If an IPP object supports more
4268 than one natural language, the object SHOULD return 'text' and 'name' values in the natural language
4269 requested where those values are generated by the Printer (see Section 3.1.4.1).

4270 For Printers that support multiple charsets and/or multiple natural languages in 'text' and 'name'
4271 attributes, different jobs may have been submitted in differing charsets and/or natural languages. All
4272 responses MUST be returned in the charset requested by the client. However, the Get-Jobs operation
4273 uses the 'textWithLanguage' and 'nameWithLanguage' mechanism to identify the differing natural
4274 languages with each job attribute returned.

4275 The Printer object also has configured charset and natural language attributes. The client can query the
4276 Printer object to determine the list of charsets and natural languages supported by the Printer object and
4277 what the Printer object's configured values are. See the "charset-configured", "charset-supported",

4278 "natural-language-configured", and "generated-natural-language-supported" Printer description attributes
4279 for more details.

4280 The "charset-supported" attribute identifies the supported charsets. If a charset is supported, the IPP
4281 object MUST be capable of converting to and from that charset into any other supported charset. In
4282 many cases, an IPP object will support only one charset and it MUST be the UTF-8 charset.

4283 The "charset-configured" attribute identifies the one supported charset which is the native charset given
4284 the current configuration of the IPP object (administrator defined).

4285 The "generated-natural-language-supported" attribute identifies the set of supported natural languages
4286 for generated messages; it is not related to the set of natural languages that must be accepted for client
4287 supplied 'text' and 'name' attributes. For client supplied 'text' and 'name' attributes, an IPP object MUST
4288 accept ALL supplied natural languages. Just because a Printer object is currently configured to support
4289 'en-us' natural language does not mean that the Printer object should reject a job if the client supplies a
4290 job name that is in 'fr-ca'.

4291 The "natural-language-configured" attribute identifies the one supported natural language for generated
4292 messages which is the native natural language given the current configuration of the IPP object
4293 (administrator defined).

4294 Attributes of type 'text' and 'name' are populated from different sources. These attributes can be
4295 categorized into following groups (depending on the source of the attribute):

- 4296 1. Some attributes are supplied by the client (e.g., the client supplied "job-name", "document-name",
4297 and "requesting-user-name" operation attributes along with the corresponding Job object's "job-
4298 name" and "job-originating-user-name" attributes). The IPP object MUST accept these attributes
4299 in any natural language no matter what the set of supported languages for generated messages
 - 4300 2. Some attributes are supplied by the system administrator (e.g., the Printer object's "printer-name"
4301 and "printer-location" attributes). These too can be in any natural language. If the natural
4302 language for these attributes is different than what a client requests, then they must be reported
4303 using the Natural Language Override mechanism.
 - 4304 3. Some attributes are supplied by the device manufacturer (e.g., the Printer object's "printer-make-
4305 and-model" attribute). These too can be in any natural language. If the natural language for
4306 these attributes is different than what a client requests, then they must be reported using the
4307 Natural Language Override mechanism.
 - 4308 4. Some attributes are supplied by the operator (e.g., the Job object's "job-message-from-operator"
4309 attribute). These too can be in any natural language. If the natural language for these attributes is
4310 different than what a client requests, then they must be reported using the Natural Language
4311 Override mechanism.
 - 4312 5. Some attributes are generated by the IPP object (e.g., the Job object's "job-state-message"
4313 attribute, the Printer object's "printer-state-message" attribute, and the "status-message" operation
4314 attribute). These attributes can only be in one of the "generated-natural-language-supported"
4315 natural languages. If a client requests some natural language for these attributes other than one of
4316 the supported values, the IPP object SHOULD respond using the value of the "natural-language-
4317 configured" attribute (using the Natural Language Override mechanism if needed).
- 4318

4319 The 'text' and 'name' attributes specified in this version of this document (additional ones will be
4320 registered according to the procedures in Section 6) are:

4321	Attributes	Source
4322	-----	-----
4323	Operation Attributes	
4324	job-name (name)	client
4325	document-name (name)	client
4326	requesting-user-name (name)	client
4327	status-message	Job or Printer object
4328		
4329	Job Template Attributes:	
4330	job-hold-until (keyword name)	client matches administrator-configured
4331	job-hold-until-default (keyword name)	client matches administrator-configured
4332	job-hold-until-supported (keyword name)	client matches administrator-configured
4333	job-sheets (keyword name)	client matches administrator-configured
4334	job-sheets-default (keyword name)	client matches administrator-configured
4335	job-sheets-supported (keyword name)	client matches administrator-configured
4336	media (keyword name)	client matches administrator-configured
4337	media-default (keyword name)	client matches administrator-configured
4338	media-supported (keyword name)	client matches administrator-configured
4339	media-ready (keyword name)	client matches administrator-configured
4340		
4341	Job Description Attributes:	
4342	job-name (name)	client or Printer object
4343	job-originating-user-name (name)	Printer object
4344	job-state-message (text)	Job or Printer object
4345	output-device-assigned (name(127))	administrator
4346	job-message-from-operator (text(127))	operator
4347		
4348	Printer Description Attributes:	
4349	printer-name (name(127))	administrator
4350	printer-location (text(127))	administrator
4351	printer-info (text(127))	administrator
4352	printer-make-and-model (text(127))	administrator or manufacturer
4353	printer-state-message (text)	Printer object
4354	printer-message-from-operator (text(127))	operator

4355 8. Security Considerations

4356 IPP objects SHOULD be deployed over protocol stacks that support the Transport Layer Security (TLS)
4357 protocol [RFC2246]. Other IPP objects MAY be deployed over protocol stacks that do not support TLS.
4358 Some IPP objects MAY be deployed over both types of protocol stacks. Those IPP objects that support
4359 TLS, are capable of supporting mutual authentication as well as privacy of messages via multiple
4360 encryption schemes. An important point about security related information for TLS access to an IPP

4361 object, is that the security-related parameters (authentication, encryption keys, etc.) are "out-of-band"
4362 to the actual IPP protocol.

4363 An IPP object that does not support TLS MAY elect to support a transport layer that provides other
4364 security mechanisms. For example, in a mapping of IPP over HTTP/1.1 [IPP-PRO], if the IPP object
4365 does not support TLS, HTTP still allows for client authentication using Digest Access Authentication
4366 (DAA) [RFC2069].

4367 It is difficult to anticipate the security risks that might exist in any given IPP environment. For example,
4368 if IPP is used within a given corporation over a private network, the risks of exposing document data
4369 may be low enough that the corporation will choose not to use encryption on that data. However, if the
4370 connection between the client and the IPP object is over a public network, the client may wish to protect
4371 the content of the information during transmission through the network with encryption.

4372 Furthermore, the value of the information being printed may vary from one IPP environment to the next.
4373 Printing payroll checks, for example, would have a different value than printing public information from
4374 a file. There is also the possibility of denial-of-service attacks, but denial-of-service attacks against
4375 printing resources are not well understood and there is no published precedents regarding this scenario.

4376 Once the authenticated identity of the requester has been supplied to the IPP object, the object uses that
4377 identity to enforce any authorization policy that might be in place. For example, one site's policy might
4378 be that only the job owner is allowed to cancel a job. The details and mechanisms to set up a particular
4379 access control policy are not part of IPP/1.1, and must be established via some other type of
4380 administrative or access control framework. However, there are operation status codes that allow an IPP
4381 server to return information back to a client about any potential access control violations for an IPP
4382 object.

4383 During a create operation, the client's identity is recorded in the Job object in an implementation-defined
4384 attribute. This information can be used to verify a client's identity for subsequent operations on that Job
4385 object in order to enforce any access control policy that might be in effect. See section 8.3 below for
4386 more details.

4387 Since the security levels or the specific threats that any given IPP system administrator may be
4388 concerned with cannot be anticipated, IPP MUST be capable of operating with different security
4389 mechanisms and security policies as required by the individual installation. Security policies might vary
4390 from very strong, to very weak, to none at all, and corresponding security mechanisms will be required.
4391 TLS supports the type of negotiated levels of security required by most, if not all, potential IPP
4392 environments. IPP environments that require no security can elect to deploy IPP objects that do not
4393 utilize the optional TLS security mechanisms.

4394 8.1 Security Scenarios

4395 The following sections describe specific security attacks for IPP environments. Where examples are
4396 provided they should be considered illustrative of the environment and not an exhaustive set. Not all of
4397 these environments will necessarily be addressed in initial implementations of IPP.

4398 8.1.1 Client and Server in the Same Security Domain

4399 This environment is typical of internal networks where traditional office workers print the output of
4400 personal productivity applications on shared work-group printers, or where batch applications print their
4401 output on large production printers. Although the identity of the user may be trusted in this environment,
4402 a user might want to protect the content of a document against such attacks as eavesdropping, replaying
4403 or tampering.

4404 8.1.2 Client and Server in Different Security Domains

4405 Examples of this environment include printing a document created by the client on a publicly available
4406 printer, such as at a commercial print shop; or printing a document remotely on a business associate's
4407 printer. This latter operation is functionally equivalent to sending the document to the business associate
4408 as a facsimile. Printing sensitive information on a Printer in a different security domain requires strong
4409 security measures. In this environment authentication of the printer is required as well as protection
4410 against unauthorized use of print resources. Since the document crosses security domains, protection
4411 against eavesdropping and document tampering are also required. It will also be important in this
4412 environment to protect Printers against "spamming" and malicious document content.

4413 8.1.3 Print by Reference

4414 When the document is not stored on the client, printing can be done by reference. That is, the print
4415 request can contain a reference, or pointer, to the document instead of the actual document itself.
4416 Standard methods currently do not exist for remote entities to "assume" the credentials of a client for
4417 forwarding requests to a 3rd party. It is anticipated that Print-By-Reference will be used to access
4418 "public" documents and that sophisticated methods for authenticating "proxies" will not be specified for
4419 version 1 of IPP.

4420 8.2 URIs for TLS and non-TLS Access

4421 As described earlier, an IPP object SHOULD support TLS access, MAY non-TLS access, or both. The
4422 "printer-uri-supported" attribute contains the Printer object's URI(s). Its companion attribute, "uri-
4423 security-supported", identifies the security mechanism used for each URI listed in the "printer-uri-
4424 supported" attribute. For each Printer operation request, a client MUST supply only one URI in the
4425 "printer-uri" operation attribute. In other words, even though the Printer supports more than one URI,
4426 the client only interacts with the Printer object using one of its URIs. This duality is not needed for Job
4427 objects, since the Printer object is the factory for Job objects, and the Printer object will generate the
4428 correct URI for new Job objects depending on the Printer object's security configuration.

4429 8.3 The "requesting-user-name" (name(MAX)) Operation Attribute

4430 Each operation MUST specify the user who is performing the operation in both of the following two
4431 ways:

- 4432 1) via the REQUIRED "requesting-user-name" operation attribute that a client SHOULD supply in
4433 all operations. The client MUST obtain the value for this attribute from an environmental or
4434 network login name for the user, rather than allowing the user to supply any value. If the client
4435 does not supply a value for "requesting-user-name", the printer MUST assume that the client is
4436 supplying some anonymous name, such as "anonymous".
- 4437 2) via an authentication mechanism of the underlying transport which may be configured to give no
4438 authentication information.
4439

4440 There are six cases to consider:

- 4441 a) the authentication mechanism gives no information, and the client doesn't specify "requesting-
4442 user-name".
- 4443 b) the authentication mechanism gives no information, but the client specifies "requesting-user-
4444 name".
- 4445 c) the authentication mechanism specifies a user which has no human readable representation, and
4446 the client doesn't specify "requesting-user-name".
- 4447 d) the authentication mechanism specifies a user which has no human readable representation, but
4448 the client specifies "requesting-user-name".
- 4449 e) the authentication mechanism specifies a user which has a human readable representation. The
4450 Printer object ignores the "requesting-user-name".
- 4451 f) the authentication mechanism specifies a user who is trusted and whose name means that the
4452 value of the "requesting-user-name", which MUST be present, is treated as the authenticated
4453 name.
4454

4455 Note: Case "f" is intended for a tightly coupled gateway and server to work together so that the "user"
4456 name is able to be that of the gateway client and not that of the gateway. Because most, if not all, system
4457 vendors will initially implement IPP via a gateway into their existing print system, this mechanism is
4458 necessary unless the authentication mechanism allows a gateway (client) to act on behalf of some other
4459 client.

4460 The user-name has two forms:

- 4461 - one that is human readable: it is held in the REQUIRED "job-originating-user-name" Job
4462 Description attribute which is set during the job creation operations. It is used for presentation
4463 only, such as returning in queries or printing on start sheets
- 4464 - one for authorization: it is held in an undefined (by IPP) Job object attribute which is set by the job
4465 creation operation. It is used to authorize other operations, such as Send-Document, Send-URI,
4466 Cancel-Job, to determine the user when the "my-jobs" attribute is specified with Get-Jobs, and to
4467 limit what attributes and values to return with Get-Job-Attributes and Get-Jobs.
4468

4469 The human readable user name:

- 4470 - is the value of the "requesting-user-name" for cases b, d and f.
4471 - comes from the authentication mechanism for case e
4472 - is some anonymous name, such as "anonymous" for cases a and c.
4473

4474 The user name used for authorization:

- 4475 - is the value of the "requesting-user-name" for cases b and f.
- 4476 - comes from the authentication mechanism for cases c, d and e
- 4477 - is some anonymous name, such as "anonymous" for case a.

4478

4479 The essence of these rules for resolving conflicting sources of user-names is that a printer
4480 implementation is free to pick either source as long as it achieves consistent results. That is, if a user
4481 uses the same path for a series of requests, the requests **MUST** appear to come from the same user from
4482 the standpoint of both the human-readable user name and the user name for authorization. This rule
4483 **MUST** continue to apply even if a request could be authenticated by two or more mechanisms. It doesn't
4484 matter which of several authentication mechanisms a Printer uses as long as it achieves consistent
4485 results. If a client uses more than one authentication mechanism, it is recommended that an
4486 administrator make all credentials resolve to the same user and user-name as much as possible.

4487 8.4 Restricted Queries

4488 In many IPP operations, a client supplies a list of attributes to be returned in the response. For security
4489 reasons, an IPP object may be configured not to return all attributes (or all values) that a client requests.
4490 The job attributes returned **MAY** depend on whether the requesting user is the same as the user that
4491 submitted the job. The IPP object **MAY** even return none of the requested attributes. In such cases, the
4492 status returned is the same as if the object had returned all requested attributes. The client cannot tell by
4493 such a response whether the requested attribute was present or absent on the object.

4494 8.5 Operations performed by operators and system administrators

4495 For the three printer operations Pause-Printer, Resume-Printer, and Purge-Jobs (see sections 3.2.7, 3.2.8
4496 and 3.2.9), the requesting user is intended to be an operator or administrator of the Printer object (see
4497 section 1). The means for authorizing an operator or administrator of the Printer object are not specified
4498 in this document.

4499 8.6 Queries on jobs submitted using non-IPP protocols

4500 If the device that an IPP Printer is representing is able to accept jobs using other job submission
4501 protocols in addition to IPP, it is **RECOMMENDED** that such an implementation at least allow such
4502 "foreign" jobs to be queried using Get-Jobs returning "job-id" and "job-uri" as 'unknown'. Such an
4503 implementation **NEED NOT** support all of the same IPP job attributes as for IPP jobs. The IPP object
4504 returns the 'unknown' out-of-band value for any requested attribute of a foreign job that is supported for
4505 IPP jobs, but not for foreign jobs.

4506 It is further **RECOMMENDED**, that the IPP Printer generate "job-id" and "job-uri" values for such
4507 "foreign jobs", if possible, so that they may be targets of other IPP operations, such as Get-Job-Attributes
4508 and Cancel-Job. Such an implementation also needs to deal with the problem of authentication of such
4509 foreign jobs. One approach would be to treat all such foreign jobs as belonging to users other than the
4510 user of the IPP client. Another approach would be for the foreign job to belong to 'anonymous'. Only if

4511 the IPP client has been authenticated as an operator or administrator of the IPP Printer object, could the
4512 foreign jobs be queried by an IPP request. Alternatively, if the security policy is to allow users to query
4513 other users' jobs, then the foreign jobs would also be visible to an end-user IPP client using Get-Jobs and
4514 Get-Job-Attributes.

4515 8.7 IPP Security Application Profile for TLS

4516 The IPP application profile for TLS follows the standard "Mandatory Cipher Suites" requirement
4517 as documented in the TLS specification [RFC2246].

4518 If a conforming IPP object supports TLS, it MUST implement and support the "Mandatory Cipher
4519 Suites" as specified in the TLS specification [RFC2246] and MAY support additional cipher suites.

4520 A conforming IPP client SHOULD support TLS including the "Mandatory Cipher Suites" as specified in
4521 the TLS specification [RFC2246]. A conforming IPP client MAY support additional cipher suites.
4522 Client implementations MUST NOT assume any other cipher suites are supported by an IPP Printer
4523 object.

4524 9. See the TLS specification [RFC2246] for a discussion of any government export restrictions on
4525 implementations conforming to the "Mandatory Cipher Suites". References

4526 [ASCII]

4527 Coded Character Set - 7-bit American Standard Code for Information Interchange (ASCII), ANSI
4528 X3.4-1986. This standard is the specification of the US-ASCII charset.

4529 [BCP-11]

4530 [Bradner S.](#), [Hovey R.](#), "The Organizations Involved in the IETF Standards Process", 1996/10/29
4531 (RFC 2028)

4532 [HTPP]

4533 J. Barnett, K. Carter, R. DeBry, "Initial Draft - Hypertext Printing Protocol - HTPP/1.0",
4534 October 1996, <ftp://ftp.pwg.org/pub/pwg/ipp/historic/http/overview.ps.gz>

4535 [IANA-CON]

4536 Narte, T. and Alvestrand, H.T.: Guidelines for Writing an IANA Considerations Section in
4537 RFCs, Work in Progress, draft-iesg-iana-considerations-04.txt, May 21, 1998.

4538 [IANA-CS]

4539 IANA Registry of Coded Character Sets: <ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets>

4540 [IANA-MT]

4541 IANA Registry of Media Types: <ftp://ftp.isi.edu/in-notes/iana/assignments/media-types/>

- 4542 [IPP-IIG]
4543 Hastings, T., Manros, C., "Internet Printing Protocol/1.1: draft-ietf-ipp-implementers-guide-v11-
4544 ???.txt, ?? 1999, work in progress.
- 4545 [IPP-IIG1.0]
4546 Hastings, T., Manros, C., "Internet Printing Protocol/1.0: Implementer's Guide", draft-ietf-ipp-
4547 implementers-guide-01.txt, February 1999, work in progress.
- 4548 [IPP LPD]
4549 Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols",
4550 draft-ietf-ipp-lpd-ipp-map-05.txt, November 1998.
- 4551 [IPP-MOD1.0]
4552 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model
4553 and Semantics", draft-ietf-ipp-model-11.txt, November, 1998.
- 4554 [IPP-PRO]
4555 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.1: Encoding and
4556 Transport", draft-ietf-ipp-protocol-v11-00.txt, February, 1999.
- 4557 [IPP-PRO1.0]
4558 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.0: Encoding and
4559 Transport", draft-ietf-ipp-protocol-07.txt, November, 1998.
- 4560 [IPP-RAT]
4561 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",
4562 draft-ietf-ipp-rat-04.txt, November, 1998.
- 4563 [IPP-REQ]
4564 Wright, D., "Design Goals for an Internet Printing Protocol", draft-ietf-ipp-req-03.txt, November,
4565 1998.
- 4566 [ISO10646-1]
4567 ISO/IEC 10646-1:1993, "Information technology -- Universal Multiple-Octet Coded Character
4568 Set (UCS) - Part 1: Architecture and Basic Multilingual Plane, JTC1/SC2."
- 4569 [ISO8859-1]
4570 ISO/IEC 8859-1:1987, "Information technology -- 8-bit One-Byte Coded Character Set - Part 1:
4571 Latin Alphabet Nr 1", 1987, JTC1/SC2.
- 4572 [ISO10175]
4573 ISO/IEC 10175 Document Printing Application (DPA), June 1996.
- 4574 [LDPA]
4575 T. Hastings, S. Isaacson, M. MacKay, C. Manros, D. Taylor, P. Zehler, "LDPA - Lightweight
4576 Document Printing Application", October 1996,
4577 <ftp://ftp.pwg.org/pub/pwg/ipp/historic/ldpa/ldpa8.pdf.gz>

- 4578 [P1387.4]
4579 Kirk, M. (editor), POSIX System Administration - Part 4: Printing Interfaces, POSIX 1387.4 D8,
4580 1994.
- 4581 [PSIS] Herriot, R. (editor), X/Open A Printing System Interoperability Specification (PSIS), August
4582 1995.
- 4583 [PWG]
4584 Printer Working Group, <http://www.pwg.org>.
- 4585 [RFC1035]
4586 P. Mockapetris, "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION", RFC
4587 1035, November 1987.
- 4588 [RFC1179]
4589 McLaughlin, L. III, (editor), "Line Printer Daemon Protocol" RFC 1179, August 1990.
- 4590 [RFC1759]
4591 Smith, R., Wright, F., Hastings, T., Zilles, S., and Gyllenskog, J., "Printer MIB", RFC 1759,
4592 March 1995.
- 4593 [RFC1766]
4594 H. Alvestrand, "Tags for the Identification of Languages", RFC 1766, March 1995.
- 4595 [RFC1903]
4596 J. Case, et.al., "Textual Conventions for Version 2 of the Simple Network Management Protocol
4597 (SNMP v2)" RFC 1903, January 1996.
- 4598 [RFC1952]
4599 P. Deutsch, "GZIP file format specification version 4.3", RFC 1952, May 1996.
- 4600 [RFC2026]
4601 S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.
- 4602 [RFC2045]
4603 N. Fried, N. Borenstein, ", Multipurpose Internet Mail Extensions (MIME) Part One: Format of
4604 Internet Message Bodies " RFC 2045, November 1996.
- 4605 [RFC2046]
4606 Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. N. Freed & N.
4607 Borenstein. November 1996. (Obsoletes RFC1521, RFC1522, RFC1590), RFC 2046.
- 4608 [RFC2048]
4609 N. Freed, J. Klensin & J. Postel, "Multipurpose Internet Mail Extension (MIME) Part Four:
4610 Registration Procedures". RFC 2048, November 1996.

- 4611 [RFC2068]
4612 R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, "Hypertext Transfer Protocol -
4613 HTTP/1.1", RFC 2068, January 1997
- 4614 [RFC2069]
4615 J. Franks, P. Hallam-Baker, J. Hostetler, P. Leach, A. Luotonen, E. Sink, L. Stewart, "An
4616 Extension to HTTP: Digest Access Authentication", RFC-2069, Jan 1997.
- 4617 [RFC2119]
4618 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 , March
4619 1997
- 4620 [RFC2228]
4621 M. Horowitz, S. Lunt, "FTP Security Extensions", RFC 2228, October 1997.
- 4622 [RFC2246]
4623 T. Dierks, C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.
- 4624 [RFC2277]
4625 H. Alvestrand, "IETF Policy on Character Sets and Languages" RFC 2277, January 1998.
- 4626 [RFC2278]
4627 N. Freed, J. Postel: "IANA CharSet Registration Procedures", RFC 2278, January 1998.
- 4628 [RFC2279]
4629 F. Yergeau , "UTF-8, a transformation format of ISO 10646", RFC 2279. January 1998.
- 4630 [RFC2316]
4631 S. Bellovin , "Report of the IAB Security Architecture Workshop", RFC 2316, April 1998.
- 4632 [RFC2396]
4633 Berners-Lee, T., Fielding, R., Masinter, L., "Uniform Resource Identifiers (URI): Generic
4634 Syntax", RFC 2396, August 1998.
- 4635 [SSL]
4636 Netscape, The SSL Protocol, Version 3, (Text version 3.02), November 1996.
- 4637 [SWP]
4638 P. Moore, B. Jahromi, S. Butler, "Simple Web Printing SWP/1.0", May 7, 1997,
4639 ftp://ftp.pwg.org/pub/pwg/ipp/new_PRO/swp9705.pdf
- 4640 10. Notices
- 4641 The IETF takes no position regarding the validity or scope of any intellectual property or other rights that
4642 might be claimed to pertain to the implementation or use of the technology described in this document

4643 or the extent to which any license under such rights might or might not be available; neither does it
4644 represent that it has made any effort to identify any such rights. Information on the IETF's procedures
4645 with respect to rights in standards-track and standards-related documentation can be found in BCP-
4646 11[BCP-11]. Copies of claims of rights made available for publication and any assurances of licenses to
4647 be made available, or the result of an attempt made to obtain a general license or permission for the use
4648 of such proprietary rights by implementers or users of this specification can be obtained from the IETF
4649 Secretariat.

4650 The IETF invites any interested party to bring to its attention any copyrights, patents or patent
4651 applications, or other proprietary rights which may cover technology that may be required to practice this
4652 standard. Please address the information to the IETF Executive Director.

4653 Copyright (C) The Internet Society (1999). All Rights Reserved.

4654 This document and translations of it may be copied and furnished to others, and derivative works that
4655 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published
4656 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright
4657 notice and this paragraph are included on all such copies and derivative works. However, this document
4658 itself may not be modified in any way, such as by removing the copyright notice or references to the
4659 Internet Society or other Internet organizations, except as needed for the purpose of developing Internet
4660 standards in which case the procedures for copyrights defined in the Internet Standards process must be
4661 followed, or as required to translate it into languages other than English.

4662 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or
4663 its successors or assigns.

4664 This document and the information contained herein is provided on an "AS IS" basis and THE
4665 INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL
4666 WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
4667 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY
4668 RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
4669 PARTICULAR PURPOSE.

4670 11. Author's Address

4671 Scott A. Isaacson (Editor)
4672 Novell, Inc.
4673 122 E 1700 S
4674 Provo, UT 84606
4675
4676 Phone: 801-861-7366
4677 Fax: 801-861-2517
4678 e-mail: sisaacson@novell.com
4679
4680 Tom Hastings

4681 Xerox Corporation
4682 737 Hawaii St. ESAE 231
4683 El Segundo, CA 90245
4684
4685 Phone: 310-333-6413
4686 Fax: 310-333-5514
4687 e-mail: hastings@cp10.es.xerox.com
4688

4689 Robert Herriot
4690 Xerox Corp.
4691 3400 Hill View Ave, Building 1
4692 Palo Alto, CA 94304
4693
4694 Phone: 650-813-7696
4695 Fax: 650-813-6860
4696 e-mail: robert.herriot@pahv.xerox.com
4697

4698 Roger deBry
4699 HUC/003G
4700 IBM Corporation
4701 P.O. Box 1900
4702 Boulder, CO 80301-9191
4703
4704 Phone: (303) 924-4080
4705 Fax: (303) 924-9889
4706 e-mail: debry@vnet.ibm.com
4707

4708 Patrick Powell
4709 Astart Technologies
4710 9475 Chesapeake Dr., Suite D
4711 San Diego, CA 95123
4712
4713 Phone: (619) 874-6543
4714 Fax: (619) 279-8424
4715 e-mail: papowell@astart.com
4716

4717 IPP Mailing List: ipp@pwg.org
4718 IPP Mailing List Subscription: ipp-request@pwg.org
4719 IPP Web Page: <http://www.pwg.org/ipp/>
4720

4721 Implementers of this specification are encouraged to join IPP Mailing List in order to participate in any
4722 discussions of clarification issues and review of registration proposals for additional attributes and
4723 values.

4724
4725 Other Participants:

4726 Chuck Adams - Tektronix
4727 Jeff Barnett - IBM
4728 Ron Bergman - Dataproducts Corp.
4729 Sylvan Butler - HP
4730 Keith Carter - IBM Corporation
4731 Jeff Copeland - QMS
4732 Andy Davidson - Tektronix
4733 Mabry Dozier - QMS
4734 Lee Farrell - Canon Information Systems
4735 Steve Gebert - IBM
4736 Babek Jahromi - Microsoft
4737 David Kellerman - Northlake Software
4738 Rick Landau - Digital
4739 Greg LeClair - Epson
4740 Harry Lewis - IBM
4741 Pete Loya - HP
4742 Ray Lutz - Cognisys
4743 Mike MacKay - Novell, Inc.
4744 Daniel Manchala - Xerox
4745 Carl-Uno Manros - Xerox
4746 Jay Martin - Underscore
4747 Larry Masinter - Xerox
4748 Stan McConnell - Xerox
4749 Ira McDonald - High North Inc.
4750 Paul Moore - Microsoft
4751 Tetsuya Morita - Ricoh
4752 Yuichi Niwa - Ricoh
4753 Pat Nogay - IBM
4754 Ron Norton - Printronics
4755 Bob Pentecost - HP
4756 Rob Rhoads - Intel
4757 Xavier Riley - Xerox
4758 David Roach - Unisys
4759 Stuart Rowley - Kyocera
4760 Hiroyuki Sato - Canon
4761 Bob Setterbo - Adobe
4762 Devon Taylor - Novell, Inc.
4763 Mike Timperman - Lexmark
4764 Randy Turner - Sharp
4765 Atsushi Yuki - Kyocera
4766 Rick Yardumian - Xerox
4767 Lloyd Young - Lexmark
4768 Bill Wagner - DPI
4769 Jim Walker - DAZEL
4770 Chris Wellens - Interworking Labs

4771 Rob Whittle - Novell, Inc.
4772 Don Wright - Lexmark
4773 Peter Zehler - Xerox
4774 Steve Zilles - Adobe

4775 12. Formats for IPP Registration Proposals

4776 In order to propose an IPP extension for registration, the proposer must submit an application to IANA
4777 by email to "iana@iana.org" or by filling out the appropriate form on the IANA web pages
4778 (<http://www.iana.org>). This section specifies the required information and the formats for proposing
4779 registrations of extensions to IPP as provided in Section 6 for:

4780

4781 1. type2 'keyword' attribute values

4782 2. type3 'keyword' attribute values

4783 3. type2 'enum' attribute values

4784 4. type3 'enum' attribute values

4785 5. attributes

4786 6. attribute syntaxes

4787 7. operations

4788 8. status codes

4789 12.1 Type2 keyword attribute values registration

4790 Type of registration: type2 keyword attribute value

4791 Name of attribute to which this keyword specification is to be added:

4792 Proposed keyword name of this keyword value:

4793 Specification of this keyword value (follow the style of IPP Model Section 4.1.2.3):

4794 Name of proposer:

4795 Address of proposer:

4796 Email address of proposer:

4797

4798 Note: For type2 keywords, the Designated Expert will be the point of contact for the approved
4799 registration specification, if any maintenance of the registration specification is needed.

4800 12.2 Type3 keyword attribute values registration

4801 Type of registration: type3 keyword attribute value

4802 Name of attribute to which this keyword specification is to be added:

4803 Proposed keyword name of this keyword value:

4804 Specification of this keyword value (follow the style of IPP Model Section 4.1.2.3):

4805 Name of proposer:

4806 Address of proposer:

4807 Email address of proposer:

4808

4809 Note: For type3 keywords, the proposer will be the point of contact for the approved registration
4810 specification, if any maintenance of the registration specification is needed.

4811 12.3 Type2 enum attribute values registration

4812 Type of registration: type2 enum attribute value

4813 Name of attribute to which this enum specification is to be added:
4814 Keyword symbolic name of this enum value:
4815 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):
4816 Specification of this enum value (follow the style of IPP Model Section 4.1.4):
4817 Name of proposer:
4818 Address of proposer:
4819 Email address of proposer:
4820
4821 Note: For type2 enums, the Designated Expert will be the point of contact for the approved registration
4822 specification, if any maintenance of the registration specification is needed.

4823 12.4 Type3 enum attribute values registration

4824 Type of registration: type3 enum attribute value
4825 Name of attribute to which this enum specification is to be added:
4826 Keyword symbolic name of this enum value:
4827 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):
4828 Specification of this enum value (follow the style of IPP Model Section 4.1.4):
4829 Name of proposer:
4830 Address of proposer:
4831 Email address of proposer:
4832
4833 Note: For type3 enums, the proposer will be the point of contact for the approved registration
4834 specification, if any maintenance of the registration specification is needed.

4835 12.5 Attribute registration

4836 Type of registration: attribute
4837 Proposed keyword name of this attribute:
4838 Types of attribute (Operation, Job Template, Job Description, Printer Description):
4839 Operations to be used with if the attribute is an operation attribute:
4840 Object (Job, Printer, etc. if bound to an object):
4841 Attribute syntax(es) (include 1setOf and range as in Section 4.2):
4842 If attribute syntax is 'keyword' or 'enum', is it type2 or type3:
4843 If this is a Printer attribute, MAY the value returned depend on "document-format" (See Section 6.2):
4844 If this is a Job Template attribute, how does its specification depend on the value of the "multiple-
4845 document-handling" attribute:
4846 Specification of this attribute (follow the style of IPP Model Section 4.2):
4847 Name of proposer:
4848 Address of proposer:
4849 Email address of proposer:
4850
4851 Note: For attributes, the IPP Designated Expert will be the point of contact for the approved registration
4852 specification, if any maintenance of the registration specification is needed.

4853 12.6 Attribute Syntax registration

4854 Type of registration: attribute syntax

4855 Proposed name of this attribute syntax:

4856 Type of attribute syntax (integer, octetString, character-string, see [IPP-PRO]):

4857 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

4858 Specification of this attribute (follow the style of IPP Model Section 4.1):

4859 Name of proposer:

4860 Address of proposer:

4861 Email address of proposer:

4862

4863 Note: For attribute syntaxes, the IPP Designated Expert will be the point of contact for the approved

4864 registration specification, if any maintenance of the registration specification is needed.

4865 12.7 Operation registration

4866 Type of registration: operation

4867 Proposed name of this operation:

4868 Numeric operation-id value (to be assigned by the IPP Designated Expert in consultation with IANA):

4869 Object Target (Job, Printer, etc. that operation is upon):

4870 Specification of this attribute (follow the style of IPP Model Section 3):

4871 Name of proposer:

4872 Address of proposer:

4873 Email address of proposer:

4874

4875 Note: For operations, the IPP Designated Expert will be the point of contact for the approved

4876 registration specification, if any maintenance of the registration specification is needed.

4877 12.8 Attribute Group registration

4878 Type of registration: attribute group

4879 Proposed name of this attribute group:

4880 Numeric tag according to [IPP-PRO] (to be assigned by the IPP Designated Expert in consultation with
4881 IANA):

4882 Operation requests and group number for each operation in which the attribute group occurs:

4883 Operation responses and group number for each operation in which the attribute group occurs:

4884 Specification of this attribute group (follow the style of IPP Model Section 3):

4885 Name of proposer:

4886 Address of proposer:

4887 Email address of proposer:

4888

4889 Note: For attribute groups, the IPP Designated Expert will be the point of contact for the approved

4890 registration specification, if any maintenance of the registration specification is needed.

4891 12.9 Status code registration

4892 Type of registration: status code

4893 Keyword symbolic name of this status code value:

4894 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

4895 Operations that this status code may be used with:

4896 Specification of this status code (follow the style of IPP Model Section 14 APPENDIX B: Status Codes
4897 and Suggested Status Code Messages):

4898 Name of proposer:

4899 Address of proposer:

4900 Email address of proposer:

4901

4902 Note: For status codes, the Designated Expert will be the point of contact for the approved registration
4903 specification, if any maintenance of the registration specification is needed.

4904 13. APPENDIX A: Terminology

4905 This specification uses the terminology defined in this section.

4906 13.1 Conformance Terminology

4907 The key words "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT",
4908 "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in
4909 RFC 2119 [RFC2119].

4910 13.1.1 NEED NOT

4911 This term is not included in RFC 2119. The verb "NEED NOT" indicates an action that the subject of
4912 the sentence does not have to implement in order to claim conformance to the standard. The verb
4913 "NEED NOT" is used instead of "MAY NOT" since "MAY NOT" sounds like a prohibition.

4914 13.2 Model Terminology

4915 13.2.1 Keyword

4916 Keywords are used within this document as identifiers of semantic entities within the abstract model (see
4917 section 4.1.2.3). Attribute names, some attribute values, attribute syntaxes, and attribute group names
4918 are represented as keywords.

4919 13.2.2 Attributes

4920 An attribute is an item of information that is associated with an instance of an IPP object. An attribute
4921 consists of an attribute name and one or more attribute values. Each attribute has a specific attribute

4922 syntax. All object attributes are defined in section 3.3.5 and all operation attributes are defined in
4923 section 3.

4924 Job Template Attributes are described in section 4.2. The client optionally supplies Job Template
4925 attributes in a create request (operation requests that create Job objects). The Printer object has
4926 associated attributes which define supported and default values for the Printer.

4927 13.2.2.1 Attribute Name

4928 Each attribute is uniquely identified in this document by its attribute name. An attribute name is a
4929 keyword. The keyword attribute name is given in the section header describing that attribute. In running
4930 text in this document, attribute names are indicated inside double quotation marks (") where the
4931 quotation marks are not part of the keyword itself.

4932 13.2.2.2 Attribute Group Name

4933 Related attributes are grouped into named groups. The name of the group is a keyword. The group
4934 name may be used in place of naming all the attributes in the group explicitly. Attribute groups are
4935 defined in section 3.

4936 13.2.2.3 Attribute Value

4937 Each attribute has one or more values. Attribute values are represented in the syntax type specified for
4938 that attribute. In running text in this document, attribute values are indicated inside single quotation
4939 marks ('), whether their attribute syntax is keyword, integer, text, etc. where the quotation marks are not
4940 part of the value itself.

4941 13.2.2.4 Attribute Syntax

4942 Each attribute is defined using an explicit syntax type. In this document, each syntax type is defined as a
4943 keyword with specific meaning. The "Encoding and Transport" document [IPP-PRO] indicates the
4944 actual "on-the-wire" encoding rules for each syntax type. Attribute syntax types are defined in section
4945 4.1.

4946 13.2.3 Supports

4947 By definition, a Printer object supports an attribute only if that Printer object responds with the
4948 corresponding attribute populated with some value(s) in a response to a query for that attribute. A
4949 Printer object supports an attribute value if the value is one of the Printer object's "supported values"
4950 attributes. The device behind a Printer object may exhibit a behavior that corresponds to some IPP
4951 attribute, but if the Printer object, when queried for that attribute, doesn't respond with the attribute, then
4952 as far as IPP is concerned, that implementation does not support that feature. If the Printer object's "xxx-
4953 supported" attribute is not populated with a particular value (even if that value is a legal value for that
4954 attribute), then that Printer object does not support that particular value.

4955 A conforming implementation **MUST** support all **REQUIRED** attributes. However, even for
4956 **REQUIRED** attributes, conformance to IPP does not mandate that all implementations support all
4957 possible values representing all possible job processing behaviors and features. For example, if a given
4958 instance of a Printer supports only certain document formats, then that Printer responds with the
4959 "document-format-supported" attribute populated with a set of values, possibly only one, taken from the
4960 entire set of possible values defined for that attribute. This limited set of values represents the Printer's
4961 set of supported document formats. Supporting an attribute and some set of values for that attribute
4962 enables IPP end users to be aware of and make use of those features associated with that attribute and
4963 those values. If an implementation chooses to not support an attribute or some specific value, then IPP
4964 end users would have no ability to make use of that feature within the context of IPP itself. However,
4965 due to existing practice and legacy systems which are not IPP aware, there might be some other
4966 mechanism outside the scope of IPP to control or request the "unsupported" feature (such as embedded
4967 instructions within the document data itself).

4968 For example, consider the "finishings-supported" attribute.

- 4969 1) If a Printer object is not physically capable of stapling, the "finishings-supported" attribute **MUST**
4970 **NOT** be populated with the value of 'staple'.
- 4971 2) A Printer object is physically capable of stapling, however an implementation chooses not to
4972 support stapling in the IPP "finishings" attribute. In this case, 'staple' **MUST NOT** be a value in
4973 the "finishings-supported" Printer object attribute. Without support for the value 'staple', an IPP
4974 end user would have no means within the protocol itself to request that a Job be stapled.
4975 However, an existing document data formatter might be able to request that the document be
4976 stapled directly with an embedded instruction within the document data. In this case, the IPP
4977 implementation does not "support" stapling, however the end user is still able to have some
4978 control over the stapling of the completed job.
- 4979 3) A Printer object is physically capable of stapling, and an implementation chooses to support
4980 stapling in the IPP "finishings" attribute. In this case, 'staple' **MUST** be a value in the "finishings-
4981 supported" Printer object attribute. Doing so, would enable end users to be aware of and make
4982 use of the stapling feature using IPP attributes.

4983
4984 Even though support for Job Template attributes by a Printer object is **OPTIONAL**, it is
4985 **RECOMMENDED** that if the device behind a Printer object is capable of realizing any feature or
4986 function that corresponds to an IPP attribute and some associated value, then that implementation
4987 **SHOULD** support that IPP attribute and value.

4988 The set of values in any of the supported value attributes is set (populated) by some administrative
4989 process or automatic sensing mechanism that is outside the scope of this IPP/1.1 document. For
4990 administrative policy and control reasons, an administrator may choose to make only a subset of possible
4991 values visible to the end user. In this case, the real output device behind the IPP Printer abstraction may
4992 be capable of a certain feature, however an administrator is specifying that access to that feature not be
4993 exposed to the end user through the IPP protocol. Also, since a Printer object may represent a logical
4994 print device (not just a physical device) the actual process for supporting a value is undefined and left up
4995 to the implementation. However, if a Printer object supports a value, some manual human action may be
4996 needed to realize the semantic action associated with the value, but no end user action is required.

4997 For example, if one of the values in the "finishings-supported" attribute is 'staple', the actual process
4998 might be an automatic staple action by a physical device controlled by some command sent to the
4999 device. Or, the actual process of stapling might be a manual action by an operator at an operator
5000 attended Printer object.

5001 For another example of how supported attributes function, consider a system administrator who desires
5002 to control all print jobs so that no job sheets are printed in order to conserve paper. To force no job
5003 sheets, the system administrator sets the only supported value for the "job-sheets-supported" attribute to
5004 'none'. In this case, if a client requests anything except 'none', the create request is rejected or the "job-
5005 sheets" value is ignored (depending on the value of "ipp-attribute-fidelity"). To force the use of job
5006 start/end sheets on all jobs, the administrator does not include the value 'none' in the "job-sheets-
5007 supported" attribute. In this case, if a client requests 'none', the create request is rejected or the "job-
5008 sheets" value is ignored (again depending on the value of "ipp-attribute-fidelity").

5009 13.2.4 print-stream page

5010 A "print-stream page" is a page according to the definition of pages in the language used to express the
5011 document data.

5012 13.2.5 impression

5013 An "impression" is the image (possibly many print-stream pages in different configurations) imposed
5014 onto a single media page.

5015 14. APPENDIX B: Status Codes and Suggested Status Code Messages

5016 This section defines status code enum keywords and values that are used to provide semantic
5017 information on the results of an operation request. Each operation response **MUST** include a status
5018 code. The response **MAY** also contain a status message that provides a short textual description of the
5019 status. The status code is intended for use by automata, and the status message is intended for the human
5020 end user. Since the status message is an **OPTIONAL** component of the operation response, an IPP
5021 application (i.e., a browser, GUI, print driver or gateway) is **NOT REQUIRED** to examine or display the
5022 status message, since it **MAY** not be returned to the application.

5023 The prefix of the status keyword defines the class of response as follows:

5024 "informational" - Request received, continuing process
5025 "successful" - The action was successfully received, understood, and accepted
5026 "redirection" - Further action must be taken in order to complete the request
5027 "client-error" - The request contains bad syntax or cannot be fulfilled
5028 "server-error" - The IPP object failed to fulfill an apparently valid request
5029

5030 As with type2 enums, IPP status codes are extensible. IPP clients are **NOT REQUIRED** to understand
5031 the meaning of all registered status codes, though such understanding is obviously desirable. However,

5032 IPP clients MUST understand the class of any status code, as indicated by the prefix, and treat any
5033 unrecognized response as being equivalent to the first status code of that class, with the exception that an
5034 unrecognized response MUST NOT be cached. For example, if an unrecognized status code of "client-
5035 error-xxx-yyy" is received by the client, it can safely assume that there was something wrong with its
5036 request and treat the response as if it had received a "client-error-bad-request" status code. In such cases,
5037 IPP applications SHOULD present the OPTIONAL message (if present) to the end user since the
5038 message is likely to contain human readable information which will help to explain the unusual status.
5039 The name of the enum is the suggested status message for US English.

5040 The status code values range from 0x0000 to 0x7FFF. The value ranges for each status code class are as
5041 follows:

5042 "successful" - 0x0000 to 0x00FF
5043 "informational" - 0x0100 to 0x01FF
5044 "redirection" - 0x0200 to 0x02FF
5045 "client-error" - 0x0400 to 0x04FF
5046 "server-error" - 0x0500 to 0x05FF
5047

5048 The top half (128 values) of each range (0x0n40 to 0x0nFF, for n = 0 to 5) is reserved for private use
5049 within each status code class. Values 0x0600 to 0x7FFF are reserved for future assignment and MUST
5050 NOT be used.

5051 14.1 Status Codes

5052 Each status code is described below. Section 14.1.5.9 contains a table that indicates which status codes
5053 apply to which operations. The Implementer's Guide [IPP-IIG] describe the suggested steps for
5054 processing IPP attributes for all operations, including returning status codes.

5055 14.1.1 Informational

5056 This class of status code indicates a provisional response and is to be used for informational purposes
5057 only.

5058 There are no status codes defined in IPP/1.1 for this class of status code.

5059 14.1.2 Successful Status Codes

5060 This class of status code indicates that the client's request was successfully received, understood, and
5061 accepted.

5062 14.1.2.1 successful-ok (0x0000)

5063 The request has succeeded and no request attributes were substituted or ignored. In the case of a
5064 response to a create request, the 'successful-ok' status code indicates that the request was successfully
5065 received and validated, and that the Job object has been created; it does not indicate that the job has been

5066 processed. The transition of the Job object into the 'completed' state is the only indicator that the job has
5067 been printed.

5068 14.1.2.2 successful-ok-ignored-or-substituted-attributes (0x0001)

5069 The request has succeeded, but some supplied (1) attributes were ignored or (2) unsupported values were
5070 substituted with supported values or were ignored in order to perform the operation without rejecting it.
5071 Unsupported attributes, attribute syntaxes, or values **MUST** be returned in the Unsupported Attributes
5072 group of the response for all operations. There is an exception to this rule for the query operations: Get-
5073 Printer-Attributes, Get-Jobs, and Get-Job-Attributes for the "requested-attributes" operation attribute
5074 only. When the supplied values of the "requested-attributes" operation attribute are requesting attributes
5075 that are not supported, the IPP object **MAY**, but is **NOT REQUIRED** to, return the "requested-attributes"
5076 attribute in the Unsupported Attribute response group (with the unsupported values only). See section
5077 3.2.1.2.

5078 14.1.2.3 successful-ok-conflicting-attributes (0x0002)

5079 The request has succeeded, but some supplied attribute values conflicted with the values of other
5080 supplied attributes. These conflicting values were either (1) substituted with (supported) values or (2)
5081 the attributes were removed in order to process the job without rejecting it. Attributes or values which
5082 conflict with other attributes and have been substituted or ignored **MUST** be returned in the Unsupported
5083 Attributes group of the response for all operations as supplied by the client. See section 3.2.1.2.

5084 14.1.3 Redirection Status Codes

5085 This class of status code indicates that further action needs to be taken to fulfill the request.

5086 There are no status codes defined in IPP/1.1 for this class of status code.

5087 14.1.4 Client Error Status Codes

5088 This class of status code is intended for cases in which the client seems to have erred. The IPP object
5089 **SHOULD** return a message containing an explanation of the error situation and whether it is a temporary
5090 or permanent condition.

5091 14.1.4.1 client-error-bad-request (0x0400)

5092 The request could not be understood by the IPP object due to malformed syntax (such as the value of a
5093 fixed length attribute whose length does not match the prescribed length for that attribute - see the
5094 Implementer's Guide [IPP-IIG]). The IPP application **SHOULD NOT** repeat the request without
5095 modifications.

5096 14.1.4.2 client-error-forbidden (0x0401)

5097 The IPP object understood the request, but is refusing to fulfill it. Additional authentication information
5098 or authorization credentials will not help and the request SHOULD NOT be repeated. This status code
5099 is commonly used when the IPP object does not wish to reveal exactly why the request has been refused
5100 or when no other response is applicable.

5101 14.1.4.3 client-error-not-authenticated (0x0402)

5102 The request requires user authentication. The IPP client may repeat the request with suitable
5103 authentication information. If the request already included authentication information, then this status
5104 code indicates that authorization has been refused for those credentials. If this response contains the
5105 same challenge as the prior response, and the user agent has already attempted authentication at least
5106 once, then the response message may contain relevant diagnostic information. This status codes reveals
5107 more information than "client-error-forbidden".

5108 14.1.4.4 client-error-not-authorized (0x0403)

5109 The requester is not authorized to perform the request. Additional authentication information or
5110 authorization credentials will not help and the request SHOULD NOT be repeated. This status code is
5111 used when the IPP object wishes to reveal that the authentication information is understandable,
5112 however, the requester is explicitly not authorized to perform the request. This status codes reveals
5113 more information than "client-error-forbidden" and "client-error-not-authenticated".

5114 14.1.4.5 client-error-not-possible (0x0404)

5115 This status code is used when the request is for something that can not happen. For example, there
5116 might be a request to cancel a job that has already been canceled or aborted by the system. The IPP
5117 client SHOULD NOT repeat the request.

5118 14.1.4.6 client-error-timeout (0x0405)

5119 The client did not produce a request within the time that the IPP object was prepared to wait. For
5120 example, a client issued a Create-Job operation and then, after a long period of time, issued a Send-
5121 Document operation and this error status code was returned in response to the Send-Document request
5122 (see section 3.3.1). The IPP object might have been forced to clean up resources that had been held for
5123 the waiting additional Documents. The IPP object was forced to close the Job since the client took too
5124 long. The client SHOULD NOT repeat the request without modifications.

5125 14.1.4.7 client-error-not-found (0x0406)

5126 The IPP object has not found anything matching the request URI. No indication is given of whether the
5127 condition is temporary or permanent. For example, a client with an old reference to a Job (a URI) tries
5128 to cancel the Job, however in the mean time the Job might have been completed and all record of it at the
5129 Printer has been deleted. This status code, 'client-error-not-found' is returned indicating that the

5130 referenced Job can not be found. This error status code is also used when a client supplies a URI as a
5131 reference to the document data in either a Print-URI or Send-URI operation, but the document can not be
5132 found.

5133 In practice, an IPP application should avoid a not found situation by first querying and presenting a list
5134 of valid Printer URIs and Job URIs to the end-user.

5135 14.1.4.8 client-error-gone (0x0407)

5136 The requested object is no longer available and no forwarding address is known. This condition should
5137 be considered permanent. Clients with link editing capabilities should delete references to the request
5138 URI after user approval. If the IPP object does not know or has no facility to determine, whether or not
5139 the condition is permanent, the status code "client-error-not-found" should be used instead.

5140 This response is primarily intended to assist the task of maintenance by notifying the recipient that the
5141 resource is intentionally unavailable and that the IPP object administrator desires that remote links to
5142 that resource be removed. It is not necessary to mark all permanently unavailable resources as "gone" or
5143 to keep the mark for any length of time -- that is left to the discretion of the IPP object administrator.

5144 14.1.4.9 client-error-request-entity-too-large (0x0408)

5145 The IPP object is refusing to process a request because the request entity is larger than the IPP object is
5146 willing or able to process. An IPP Printer returns this status code when it limits the size of print jobs and
5147 it receives a print job that exceeds that limit or when the attributes are so many that their encoding
5148 causes the request entity to exceed IPP object capacity.

5149 14.1.4.10 client-error-request-value-too-long (0x0409)

5150 The IPP object is refusing to service the request because one or more of the client-supplied attributes has
5151 a variable length value that is longer than the maximum length specified for that attribute. The IPP
5152 object might not have sufficient resources (memory, buffers, etc.) to process (even temporarily),
5153 interpret, and/or ignore a value larger than the maximum length. Another use of this error code is when
5154 the IPP object supports the processing of a large value that is less than the maximum length, but during
5155 the processing of the request as a whole, the object may pass the value onto some other system
5156 component which is not able to accept the large value. For more details, see the Implementer's Guide
5157 [IPP-IIG] .

5158 Note: For attribute values that are URIs, this rare condition is only likely to occur when a client has
5159 improperly submitted a request with long query information (e.g. an IPP application allows an end-user
5160 to enter an invalid URI), when the client has descended into a URI "black hole" of redirection (e.g., a
5161 redirected URI prefix that points to a suffix of itself), or when the IPP object is under attack by a client
5162 attempting to exploit security holes present in some IPP objects using fixed-length buffers for reading or
5163 manipulating the Request-URI.

5164 14.1.4.11 client-error-document-format-not-supported (0x040A)

5165 The IPP object is refusing to service the request because the document data is in a format, as specified in
5166 the "document-format" operation attribute, that is not supported by the Printer object. This error is
5167 returned independent of the client-supplied "ipp-attribute-fidelity". The Printer object MUST return this
5168 status code, even if there are other attributes that are not supported as well, since this error is a bigger
5169 problem than with Job Template attributes.

5170 14.1.4.12 client-error-attributes-or-values-not-supported (0x040B)

5171 In a create request, if the Printer object does not support one or more attributes, attribute syntaxes, or
5172 attribute values supplied in the request and the client supplied the "ipp-attributes-fidelity" operation
5173 attribute with the 'true' value, the Printer object MUST return this status code. For example, if the
5174 request indicates 'iso-a4' media, but that media type is not supported by the Printer object. Or, if the
5175 client supplies an optional attribute and the attribute itself is not even supported by the Printer. If the
5176 "ipp-attribute-fidelity" attribute is 'false', the Printer MUST ignore or substitute values for unsupported
5177 attributes and values rather than reject the request and return this status code.

5178 For any operation where a client requests attributes (such as a Get-Jobs, Get-Printer-Attributes, or Get-
5179 Job-Attributes operation), if the IPP object does not support one or more of the requested attributes, the
5180 IPP object simply ignores the unsupported requested attributes and processes the request as if they had
5181 not been supplied, rather than returning this status code. In this case, the IPP object MUST return the
5182 'successful-ok-ignored-or-substituted-attributes' status code and MAY return the unsupported attributes
5183 as values of the "requested-attributes" in the Unsupported Attributes Group (see section 14.1.2.2).

5184 14.1.4.13 client-error-uri-scheme-not-supported (0x040C)

5185 The type of the client supplied URI in a Print-URI or a Send-URI operation is not supported.

5186 14.1.4.14 client-error-charset-not-supported (0x040D)

5187 For any operation, if the IPP Printer does not support the charset supplied by the client in the "attributes-
5188 charset" operation attribute, the Printer MUST reject the operation and return this status and any 'text' or
5189 'name' attributes using the 'utf-8' charset (see Section 3.1.4.1).

5190 14.1.4.15 client-error-conflicting-attributes (0x040E)

5191 The request is rejected because some attribute values conflicted with the values of other attributes which
5192 this specification does not permit to be substituted or ignored.

5193 14.1.4.16 client-error-compression-not-supported (0x040F)

5194 The IPP object is refusing to service the request because the document data, as specified in the
5195 "compression" operation attribute, is compressed in a way that is not supported by the Printer object.
5196 This error is returned independent of the client-supplied "ipp-attribute-fidelity". The Printer object

5197 MUST return this status code, even if there are other attributes that are not supported as well, since this
5198 error is a bigger problem than with Job Template attributes. **Issue 6.**

5199

5200 14.1.5 Server Error Status Codes

5201 This class of status codes indicates cases in which the IPP object is aware that it has erred or is incapable
5202 of performing the request. The IPP object SHOULD include a message containing an explanation of the
5203 error situation, and whether it is a temporary or permanent condition.

5204 14.1.5.1 server-error-internal-error (0x0500)

5205 The IPP object encountered an unexpected condition that prevented it from fulfilling the request. This
5206 error status code differs from "server-error-temporary-error" in that it implies a more permanent type of
5207 internal error. It also differs from "server-error-device-error" in that it implies an unexpected condition
5208 (unlike a paper-jam or out-of-toner problem which is undesirable but expected). This error status code
5209 indicates that probably some knowledgeable human intervention is required.

5210 14.1.5.2 server-error-operation-not-supported (0x0501)

5211 The IPP object does not support the functionality required to fulfill the request. This is the appropriate
5212 response when the IPP object does not recognize an operation or is not capable of supporting it.

5213 14.1.5.3 server-error-service-unavailable (0x0502)

5214 The IPP object is currently unable to handle the request due to a temporary overloading or maintenance
5215 of the IPP object. The implication is that this is a temporary condition which will be alleviated after
5216 some delay. If known, the length of the delay may be indicated in the message. If no delay is given, the
5217 IPP application should handle the response as it would for a "server-error-temporary-error" response. If
5218 the condition is more permanent, the error status codes "client-error-gone" or "client-error-not-found"
5219 could be used.

5220 14.1.5.4 server-error-version-not-supported (0x0503)

5221 The IPP object does not support, or refuses to support, the IPP protocol version that was used in the
5222 request message. The IPP object is indicating that it is unable or unwilling to complete the request using
5223 the same version as supplied in the request other than with this error message. The response should
5224 contain a Message describing why that version is not supported and what other versions are supported by
5225 that IPP object.

5226 A conforming IPP/1.1 client MUST specify a valid version ('1.1' or '1.0') on each request. A conforming
5227 IPP/1.1 object MUST NOT return this status code to a conforming IPP/1.1 or IPP/1.0 client. An IPP
5228 object MUST return this status code to a non-conforming IPP client. The response MUST identify in the

5229 "version-number" operation attribute the closest version number that the IPP object does support. For
5230 example, if a client supplies version '1.0', a conforming IPP/1.1 object MUST respond with version '1.0'.

5231 14.1.5.5 server-error-device-error (0x0504)

5232 A printer error, such as a paper jam, occurs while the IPP object processes a Print or Send operation.
5233 The response contains the true Job Status (the values of the "job-state" and "job-state-reasons"
5234 attributes). Additional information can be returned in the optional "job-state-message" attribute value or
5235 in the OPTIONAL status message that describes the error in more detail. This error status code is only
5236 returned in situations where the Printer is unable to accept the create request because of such a device
5237 error. For example, if the Printer is unable to spool, and can only accept one job at a time, the reason it
5238 might reject a create request is that the printer currently has a paper jam. In many cases however, where
5239 the Printer object can accept the request even though the Printer has some error condition, the
5240 'successful-ok' status code will be returned. In such a case, the client would look at the returned Job
5241 Object Attributes or later query the Printer to determine its state and state reasons.

5242 14.1.5.6 server-error-temporary-error (0x0505)

5243 A temporary error such as a buffer full write error, a memory overflow (i.e. the document data exceeds
5244 the memory of the Printer), or a disk full condition, occurs while the IPP Printer processes an operation.
5245 The client MAY try the unmodified request again at some later point in time with an expectation that the
5246 temporary internal error condition may have been cleared. Alternatively, as an implementation option, a
5247 Printer object MAY delay the response until the temporary condition is cleared so that no error is
5248 returned.

5249 14.1.5.7 server-error-not-accepting-jobs (0x0506)

5250 A temporary error indicating that the Printer is not currently accepting jobs, because the administrator
5251 has set the value of the Printer's "printer-is-not-accepting-jobs" attribute to 'false' (by means outside the
5252 scope of this IPP/1.1 document).

5253 14.1.5.8 server-error-busy (0x0507)

5254 A temporary error indicating that the Printer is too busy processing jobs and/or other requests. The client
5255 SHOULD try the unmodified request again at some later point in time with an expectation that the
5256 temporary busy condition will have been cleared.

5257 14.1.5.9 server-error-job-canceled (0x0508)

5258 An error indicating that the job has been canceled by an operator or the system while the client was
5259 transmitting the data to the IPP Printer. If a job-id and job-uri had been created, then they are returned in
5260 the Print-Job, Send-Document, or Send-URI response as usual; otherwise, no job-id and job-uri are
5261 returned in the response.

5262 14.2 Status Codes for IPP Operations

5263 PJ = Print-Job, PU = Print-URI, CJ = Create-Job, SD = Send-Document
 5264 SU = Send-URI, V = Validate-Job, GA = Get-Job-Attributes and
 5265 Get-Printer-Attributes, GJ = Get-Jobs, C = Cancel-Job

5266		IPP Operations								
5267	IPP Status Keyword	PJ	PU	CJ	SD	SU	V	GA	GJ	C
5268	-----	--	--	--	--	--	--	--	--	--
5269	successful-ok	x	x	x	x	x	x	x	x	x
5270	successful-ok-ignored-or-substituted-	x	x	x	x	x	x	x	x	x
5271	attributes									
5272	successful-ok-conflicting-attributes	x	x	x	x	x	x	x	x	x
5273	client-error-bad-request	x	x	x	x	x	x	x	x	x
5274	client-error-forbidden	x	x	x	x	x	x	x	x	x
5275	client-error-not-authenticated	x	x	x	x	x	x	x	x	x
5276	client-error-not-authorized	x	x	x	x	x	x	x	x	x
5277	client-error-not-possible	x	x	x	x	x	x	x	x	x
5278	client-error-timeout				x	x				
5279	client-error-not-found	x	x	x	x	x	x	x	x	x
5280	client-error-gone	x	x	x	x	x	x	x	x	x
5281	client-error-request-entity-too-large	x	x	x	x	x	x	x	x	x
5282	client-error-request-value-too-long	x	x	x	x	x	x	x	x	x
5283	client-error-document-format-not-	x	x		x	x	x	x		
5284	supported									
5285	client-error-attributes-or-values-not-	x	x	x	x	x	x	x	x	x
5286	supported									
5287	client-error-uri-scheme-not-supported		x			x				
5288	client-error-charset-not-supported	x	x	x	x	x	x	x	x	x
5289	client-error-conflicting-attributes	x	x	x	x	x	x	x	x	x
5290	server-error-internal-error	x	x	x	x	x	x	x	x	x
5291	server-error-operation-not-supported		x	x	x	x				
5292	server-error-service-unavailable	x	x	x	x	x	x	x	x	x
5293	server-error-version-not-supported	x	x	x	x	x	x	x	x	x
5294	server-error-device-error	x	x	x	x	x				
5295	server-error-temporary-error	x	x	x	x	x				
5296	server-error-not-accepting-jobs	x	x	x			x			
5297	server-error-busy	x	x	x	x	x	x	x	x	x
5298	server-error-job-canceled	x			x					
5299										
5300										

5301	HJ = Hold-Job, RJ = Release-Job, RS = Restart-Job						
5302	PP = Pause-Printer, RP = Resume-Printer, PJ = Purge-Jobs						
5303							
5304		IPP Operations (cont.)					
5305	IPP Status Keyword	HJ	RJ	RS	PP	RP	PJ
5306	-----	--	--	--	--	--	--
5307	successful-ok	x	x	x	x	x	x
5308	successful-ok-ignored-or-substituted-	x	x	x	x	x	x
5309	attributes						
5310	successful-ok-conflicting-attributes	x	x	x	x	x	x
5311	client-error-bad-request	x	x	x	x	x	x
5312	client-error-forbidden	x	x	x	x	x	x
5313	client-error-not-authenticated	x	x	x	x	x	x
5314	client-error-not-authorized	x	x	x	x	x	x
5315	client-error-not-possible	x	x	x	x	x	x
5316	client-error-timeout						
5317	client-error-not-found	x	x	x	x	x	x
5318	client-error-gone	x	x	x	x	x	x
5319	client-error-request-entity-too-large	x	x	x	x	x	x
5320	client-error-request-value-too-long	x	x	x	x	x	x
5321	client-error-document-format-not-						
5322	supported						
5323	client-error-attributes-or-values-not-	x	x	x	x	x	x
5324	supported						
5325	client-error-uri-scheme-not-supported						
5326	client-error-charset-not-supported	x	x	x	x	x	x
5327	client-error-conflicting-attributes	x	x	x	x	x	x
5328	server-error-internal-error	x	x	x	x	x	x
5329	server-error-operation-not-supported	x	x	x	x	x	x
5330	server-error-service-unavailable	x	x	x	x	x	x
5331	server-error-version-not-supported	x	x	x	x	x	x
5332	server-error-device-error						
5333	server-error-temporary-error						
5334	server-error-not-accepting-jobs						
5335	server-error-busy	x	x	x	x	x	x
5336	server-error-job-canceled						

5337

5338 15. APPENDIX C: "media" keyword values

5339 Standard keyword values are taken from several sources.

5340 Standard values are defined (taken from DPA[ISO10175] and the Printer MIB[RFC1759]):

- 5341 'default': The default medium for the output device
- 5342 'iso-a4-white': Specifies the ISO A4 white medium
- 5343 'iso-a4-colored': Specifies the ISO A4 colored medium
- 5344 'iso-a4-transparent': Specifies the ISO A4 transparent medium
- 5345 'iso-a3-white': Specifies the ISO A3 white medium
- 5346 'iso-a3-colored': Specifies the ISO A3 colored medium
- 5347 'iso-a5-white': Specifies the ISO A5 white medium
- 5348 'iso-a5-colored': Specifies the ISO A5 colored medium
- 5349 'iso-b4-white': Specifies the ISO B4 white medium
- 5350 'iso-b4-colored': Specifies the ISO B4 colored medium
- 5351 'iso-b5-white': Specifies the ISO B5 white medium
- 5352 'iso-b5-colored': Specifies the ISO B5 colored medium
- 5353 'jis-b4-white': Specifies the JIS B4 white medium
- 5354 'jis-b4-colored': Specifies the JIS B4 colored medium
- 5355 'jis-b5-white': Specifies the JIS B5 white medium
- 5356 'jis-b5-colored': Specifies the JIS B5 colored medium

5357

5358 The following standard values are defined for North American media:

- 5359 'na-letter-white': Specifies the North American letter white medium
- 5360 'na-letter-colored': Specifies the North American letter colored medium
- 5361 'na-letter-transparent': Specifies the North American letter transparent medium
- 5362 'na-legal-white': Specifies the North American legal white medium
- 5363 'na-legal-colored': Specifies the North American legal colored medium

5364

5365 The following standard values are defined for envelopes:

- 5366 'iso-b4-envelope': Specifies the ISO B4 envelope medium
- 5367 'iso-b5-envelope': Specifies the ISO B5 envelope medium
- 5368 'iso-c3-envelope': Specifies the ISO C3 envelope medium
- 5369 'iso-c4-envelope': Specifies the ISO C4 envelope medium
- 5370 'iso-c5-envelope': Specifies the ISO C5 envelope medium
- 5371 'iso-c6-envelope': Specifies the ISO C6 envelope medium
- 5372 'iso-designated-long-envelope': Specifies the ISO Designated Long envelope medium
- 5373 'na-10x13-envelope': Specifies the North American 10x13 envelope medium
- 5374 'na-9x12-envelope': Specifies the North American 9x12 envelope medium

5375 'monarch-envelope': Specifies the Monarch envelope
5376 'na-number-10-envelope': Specifies the North American number 10 business envelope medium
5377 'na-7x9-envelope': Specifies the North American 7x9 inch envelope
5378 'na-9x11-envelope': Specifies the North American 9x11 inch envelope
5379 'na-10x14-envelope': Specifies the North American 10x14 inch envelope
5380 'na-number-9-envelope': Specifies the North American number 9 business envelope
5381 'na-6x9-envelope': Specifies the North American 6x9 inch envelope
5382 'na-10x15-envelope': Specifies the North American 10x15 inch envelope
5383

5384 The following standard values are defined for the less commonly used media (white-only):

5385 'executive-white': Specifies the white executive medium
5386 'folio-white': Specifies the folio white medium
5387 'invoice-white': Specifies the white invoice medium
5388 'ledger-white': Specifies the white ledger medium
5389 'quarto-white': Specifies the white quarto medium
5390 'iso-a0-white': Specifies the ISO A0 white medium
5391 'iso-a1-white': Specifies the ISO A1 white medium
5392 'iso-a2-white': Specifies the ISO A2 white medium
5393 'iso-a6-white': Specifies the ISO A6 white medium
5394 'iso-a7-white': Specifies the ISO A7 white medium
5395 'iso-a8-white': Specifies the ISO A8 white medium
5396 'iso-a9-white': Specifies the ISO A9 white medium
5397 'iso-10-white': Specifies the ISO A10 white medium
5398 'iso-b0-white': Specifies the ISO B0 white medium
5399 'iso-b1-white': Specifies the ISO B1 white medium
5400 'iso-b2-white': Specifies the ISO B2 white medium
5401 'iso-b3-white': Specifies the ISO B3 white medium
5402 'iso-b6-white': Specifies the ISO B6 white medium
5403 'iso-b7-white': Specifies the ISO B7 white medium
5404 'iso-b8-white': Specifies the ISO B8 white medium
5405 'iso-b9-white': Specifies the ISO B9 white medium
5406 'iso-b10-white': Specifies the ISO B10 white medium
5407 'jis-b0-white': Specifies the JIS B0 white medium
5408 'jis-b1-white': Specifies the JIS B1 white medium
5409 'jis-b2-white': Specifies the JIS B2 white medium
5410 'jis-b3-white': Specifies the JIS B3 white medium
5411 'jis-b6-white': Specifies the JIS B6 white medium
5412 'jis-b7-white': Specifies the JIS B7 white medium
5413 'jis-b8-white': Specifies the JIS B8 white medium
5414 'jis-b9-white': Specifies the JIS B9 white medium
5415 'jis-b10-white': Specifies the JIS B10 white medium
5416

5417 The following standard values are defined for engineering media:

5418 'a': Specifies the engineering A size medium
5419 'b': Specifies the engineering B size medium
5420 'c': Specifies the engineering C size medium
5421 'd': Specifies the engineering D size medium
5422 'e': Specifies the engineering E size medium
5423

5424 The following standard values are defined for input-trays (from ISO DPA and the Printer MIB):

5425 'top': The top input tray in the printer.
5426 'middle': The middle input tray in the printer.
5427 'bottom': The bottom input tray in the printer.
5428 'envelope': The envelope input tray in the printer.
5429 'manual': The manual feed input tray in the printer.
5430 'large-capacity': The large capacity input tray in the printer.
5431 'main': The main input tray
5432 'side': The side input tray
5433

5434 The following standard values are defined for media sizes (from ISO DPA):

5435 'iso-a0': Specifies the ISO A0 size: 841 mm by 1189 mm as defined in ISO 216
5436 'iso-a1': Specifies the ISO A1 size: 594 mm by 841 mm as defined in ISO 216
5437 'iso-a2': Specifies the ISO A2 size: 420 mm by 594 mm as defined in ISO 216
5438 'iso-a3': Specifies the ISO A3 size: 297 mm by 420 mm as defined in ISO 216
5439 'iso-a4': Specifies the ISO A4 size: 210 mm by 297 mm as defined in ISO 216
5440 'iso-a5': Specifies the ISO A5 size: 148 mm by 210 mm as defined in ISO 216
5441 'iso-a6': Specifies the ISO A6 size: 105 mm by 148 mm as defined in ISO 216
5442 'iso-a7': Specifies the ISO A7 size: 74 mm by 105 mm as defined in ISO 216
5443 'iso-a8': Specifies the ISO A8 size: 52 mm by 74 mm as defined in ISO 216
5444 'iso-a9': Specifies the ISO A9 size: 37 mm by 52 mm as defined in ISO 216
5445 'iso-a10': Specifies the ISO A10 size: 26 mm by 37 mm as defined in ISO 216
5446 'iso-b0': Specifies the ISO B0 size: 1000 mm by 1414 mm as defined in ISO 216
5447 'iso-b1': Specifies the ISO B1 size: 707 mm by 1000 mm as defined in ISO 216
5448 'iso-b2': Specifies the ISO B2 size: 500 mm by 707 mm as defined in ISO 216
5449 'iso-b3': Specifies the ISO B3 size: 353 mm by 500 mm as defined in ISO 216
5450 'iso-b4': Specifies the ISO B4 size: 250 mm by 353 mm as defined in ISO 216
5451 'iso-b5': Specifies the ISO B5 size: 176 mm by 250 mm as defined in ISO 216
5452 'iso-b6': Specifies the ISO B6 size: 125 mm by 176 mm as defined in ISO 216
5453 'iso-b7': Specifies the ISO B7 size: 88 mm by 125 mm as defined in ISO 216
5454 'iso-b8': Specifies the ISO B8 size: 62 mm by 88 mm as defined in ISO 216
5455 'iso-b9': Specifies the ISO B9 size: 44 mm by 62 mm as defined in ISO 216
5456 'iso-b10': Specifies the ISO B10 size: 31 mm by 44 mm as defined in ISO 216
5457 'na-letter': Specifies the North American letter size: 8.5 inches by 11 inches
5458 'na-legal': Specifies the North American legal size: 8.5 inches by 14 inches
5459 'executive': Specifies the executive size (7.25 X 10.5 in)
5460 'folio': Specifies the folio size (8.5 X 13 in)

5461 'invoice': Specifies the invoice size (5.5 X 8.5 in)
5462 'ledger': Specifies the ledger size (11 X 17 in)
5463 'quarto': Specifies the quarto size (8.5 X 10.83 in)
5464 'iso-c3': Specifies the ISO C3 size: 324 mm by 458 mm as defined in ISO 269
5465 'iso-c4': Specifies the ISO C4 size: 229 mm by 324 mm as defined in ISO 269
5466 'iso-c5': Specifies the ISO C5 size: 162 mm by 229 mm as defined in ISO 269
5467 'iso-c6': Specifies the ISO C6 size: 114 mm by 162 mm as defined in ISO 269
5468 'iso-designated-long': Specifies the ISO Designated Long size: 110 mm by 220 mm as defined in ISO
5469 269
5470 'na-10x13-envelope': Specifies the North American 10x13 size: 10 inches by 13 inches
5471 'na-9x12-envelope': Specifies the North American 9x12 size: 9 inches by 12 inches
5472 'na-number-10-envelope': Specifies the North American number 10 business envelope size: 4.125
5473 inches by 9.5 inches
5474 'na-7x9-envelope': Specifies the North American 7x9 inch envelope size
5475 'na-9x11-envelope': Specifies the North American 9x11 inch envelope size
5476 'na-10x14-envelope': Specifies the North American 10x14 inch envelope size
5477 'na-number-9-envelope': Specifies the North American number 9 business envelope size
5478 'na-6x9-envelope': Specifies the North American 6x9 envelope size
5479 'na-10x15-envelope': Specifies the North American 10x15 envelope size
5480 'monarch-envelope': Specifies the Monarch envelope size (3.87 x 7.5 in)
5481 'jis-b0': Specifies the JIS B0 size: 1030mm x 1456mm
5482 'jis-b1': Specifies the JIS B1 size: 728mm x 1030mm
5483 'jis-b2': Specifies the JIS B2 size: 515mm x 728mm
5484 'jis-b3': Specifies the JIS B3 size: 364mm x 515mm
5485 'jis-b4': Specifies the JIS B4 size: 257mm x 364mm
5486 'jis-b5': Specifies the JIS B5 size: 182mm x 257mm
5487 'jis-b6': Specifies the JIS B6 size: 128mm x 182mm
5488 'jis-b7': Specifies the JIS B7 size: 91mm x 128mm
5489 'jis-b8': Specifies the JIS B8 size: 64mm x 91mm
5490 'jis-b9': Specifies the JIS B9 size: 45mm x 64mm
5491 'jis-b10': Specifies the JIS B10 size: 32mm x 45mm

5492 16. APPENDIX D: Processing IPP Attributes

5493 When submitting a print job to a Printer object, the IPP model allows a client to supply operation and
5494 Job Template attributes along with the document data. These Job Template attributes in the create
5495 request affect the rendering, production and finishing of the documents in the job. Similar types of
5496 instructions may also be contained in the document to be printed, that is, embedded within the print data
5497 itself. In addition, the Printer has a set of attributes that describe what rendering and finishing options
5498 which are supported by that Printer. This model, which allows for flexibility and power, also introduces
5499 the potential that at job submission time, these client-supplied attributes may conflict with either:

- 5500 - what the implementation is capable of realizing (i.e., what the Printer supports), as well as
- 5501 - the instructions embedded within the print data itself.

5502

5503 The following sections describe how these two types of conflicts are handled in the IPP model.

5504 16.1 Fidelity

5505 If there is a conflict between what the client requests and what a Printer object supports, the client may
5506 request one of two possible conflict handling mechanisms:

- 5507 1) either reject the job since the job can not be processed exactly as specified, or
 - 5508 2) allow the Printer to make any changes necessary to proceed with processing the Job the best it can.
- 5509

5510 In the first case the client is indicating to the Printer object: "Print the job exactly as specified with no
5511 exceptions, and if that can't be done, don't even bother printing the job at all." In the second case, the
5512 client is indicating to the Printer object: "It is more important to make sure the job is printed rather than
5513 be processed exactly as specified; just make sure the job is printed even if client supplied attributes need
5514 to be changed or ignored."

5515 The IPP model accounts for this situation by introducing an "ipp-attribute-fidelity" attribute.

5516 In a create request, "ipp-attribute-fidelity" is a boolean operation attribute that is **OPTIONALLY**
5517 supplied by the client. The value 'true' indicates that total fidelity to client supplied Job Template
5518 attributes and values is required. The client is requesting that the Job be printed exactly as specified, and
5519 if that is not possible then the job **MUST** be rejected rather than processed incorrectly. The value 'false'
5520 indicates that a reasonable attempt to print the Job is acceptable. If a Printer does not support some of
5521 the client supplied Job Template attributes or values, the Printer **MUST** ignore them or substitute any
5522 supported value for unsupported values, respectively. The Printer may choose to substitute the default
5523 value associated with that attribute, or use some other supported value that is similar to the unsupported
5524 requested value. For example, if a client supplies a "media" value of 'na-letter', the Printer may choose
5525 to substitute 'iso-a4' rather than a default value of 'envelope'. If the client does not supply the "ipp-
5526 attribute-fidelity" attribute, the Printer assumes a value of 'false'.

5527 Each Printer implementation **MUST** support both types of "fidelity" printing (that is whether the client
5528 supplies a value of 'true' or 'false'):

- 5529 - If the client supplies 'false' or does not supply the attribute, the Printer object **MUST** always accept
5530 the request by ignoring unsupported Job Template attributes and by substituting unsupported
5531 values of supported Job Template attributes with supported values.
 - 5532 - If the client supplies 'true', the Printer object **MUST** reject the request if the client supplies
5533 unsupported Job Template attributes.
- 5534

5535 Since a client can always query a Printer to find out exactly what is and is not supported, "ipp-attribute-
5536 fidelity" set to 'false' is useful when:

- 5537 1) The End-User uses a command line interface to request attributes that might not be supported.
- 5538 2) In a GUI context, if the End User expects the job might be moved to another printer and prefers a
5539 sub-optimal result to nothing at all.
- 5540 3) The End User just wants something reasonable in lieu of nothing at all.

5541

5542 16.2 Page Description Language (PDL) Override

5543 If there is a conflict between the value of an IPP Job Template attribute and a corresponding instruction
5544 in the document data, the value of the IPP attribute SHOULD take precedence over the document
5545 instruction. Consider the case where a previously formatted file of document data is sent to an IPP
5546 Printer. In this case, if the client supplies any attributes at job submission time, the client desires that
5547 those attributes override the embedded instructions. Consider the case were a previously formatted
5548 document has embedded in it commands to load 'iso-a4' media. However, the document is passed to an
5549 end user that only has access to a printer with 'na-letter' media loaded. That end user most likely wants
5550 to submit that document to an IPP Printer with the "media" Job Template attribute set to 'na-letter'. The
5551 job submission attribute should take precedence over the embedded PDL instruction. However, until
5552 companies that supply document data interpreters allow a way for external IPP attributes to take
5553 precedence over embedded job production instructions, a Printer might not be able to support the
5554 semantics that IPP attributes override the embedded instructions.

5555 The IPP model accounts for this situation by introducing a "pdl-override-supported" attribute that
5556 describes the Printer objects capabilities to override instructions embedded in the PDL data stream. The
5557 value of the "pdl-override-supported" attribute is configured by means outside the scope of this IPP/1.1
5558 document.

5559 This REQUIRED Printer attribute takes on the following values:

- 5560 - 'attempted': This value indicates that the Printer object attempts to make the IPP attribute values
5561 take precedence over embedded instructions in the document data, however there is no guarantee.
 - 5562 - 'not-attempted': This value indicates that the Printer object makes no attempt to make the IPP
5563 attribute values take precedence over embedded instructions in the document data.
- 5564

5565 At job processing time, an implementation that supports the value of 'attempted' might do one of several
5566 different actions:

- 5567 1) Generate an output device specific command sequence to realize the feature represented by the
5568 IPP attribute value.
 - 5569 2) Parse the document data itself and replace the conflicting embedded instruction with a new
5570 embedded instruction that matches the intent of the IPP attribute value.
 - 5571 3) Indicate to the Printer that external supplied attributes take precedence over embedded instructions
5572 and then pass the external IPP attribute values to the document data interpreter.
 - 5573 4) Anything else that allows for the semantics that IPP attributes override embedded document data
5574 instructions.
- 5575

5576 Since 'attempted' does not offer any type of guarantee, even though a given Printer object might not do a
5577 very "good" job of attempting to ensure that IPP attributes take a higher precedence over instructions
5578 embedded in the document data, it would still be a conforming implementation.

5579 At job processing time, an implementation that supports the value of 'not-attempted' might do one of the
5580 following actions:

- 5581 1) Simply pre-pend the document data with the PDL instruction that corresponds to the client-
5582 supplied PDL attribute, such that if the document data also has the same PDL instruction, it will
5583 override what the Printer object pre-pended. In other words, this implementation is using the
5584 same implementation semantics for the client-supplied IPP attributes as for the Printer object
5585 defaults.
- 5586 2) Parse the document data and replace the conflicting embedded instruction with a new embedded
5587 instruction that approximates, but does not match, the semantic intent of the IPP attribute value.
5588

5589 Note: The "ipp-attribute-fidelity" attribute applies to the Printer's ability to either accept or reject other
5590 unsupported Job Template attributes. In other words, if "ipp-attribute-fidelity" is set to 'true', a Job is
5591 accepted if and only if the client supplied Job Template attributes and values are supported by the
5592 Printer. Whether these attributes actually affect the processing of the Job when the document data
5593 contains embedded instructions depends on the ability of the Printer to override the instructions
5594 embedded in the document data with the semantics of the IPP attributes. If the document data attributes
5595 can be overridden ("pdl-override-supported" set to 'attempted'), the Printer makes an attempt to use the
5596 IPP attributes when processing the Job. If the document data attributes can not be overridden ("pdl-
5597 override-supported" set to 'not-attempted'), the Printer makes no attempt to override the embedded
5598 document data instructions with the IPP attributes when processing the Job, and hence, the IPP attributes
5599 may fail to affect the Job processing and output when the corresponding instruction is embedded in the
5600 document data.

5601 16.3 Using Job Template Attributes During Document Processing.

5602 The Printer object uses some of the Job object's Job Template attributes during the processing of the
5603 document data associated with that job. These include, but are not limited to, "orientation-requested",
5604 "number-up", "sides", "media", and "copies". The processing of each document in a Job Object MUST
5605 follow the steps below. These steps are intended only to identify when and how attributes are to be used
5606 in processing document data and any alternative steps that accomplishes the same effect can be used to
5607 implement this specification.

- 5608 1. Using the client supplied "document-format" attribute or some form of document format detection
5609 algorithm (if the value of "document-format" is not specific enough), determine whether or not
5610 the document data has already been formatted for printing. If the document data has been
5611 formatted, then go to step 2. Otherwise, the document data MUST be formatted. The formatting
5612 detection algorithm is implementation defined and is not specified by this specification. The
5613 formatting of the document data uses the "orientation-requested" attribute to determine how the
5614 formatted print data should be placed on a print-stream page, see section 4.2.10 for the details.
5615
- 5616 2. The document data is in the form of a print-stream in a known media type. The "page-ranges"
5617 attribute is used to select, as specified in section 4.2.7, a sub-sequence of the pages in the print-
5618 stream that are to be processed and images.
5619

5620 3. The input to this step is a sequence of print-stream pages. This step is controlled by the "number-
5621 up" attribute. If the value of "number-up" is N, then during the processing of the print-stream
5622 pages, each N print-stream pages are positioned, as specified in section 4.2.9, to create a single
5623 impression. If a given document does not have N more print-stream pages, then the completion
5624 of the impression is controlled by the "multiple-document-handling" attribute as described in
5625 section 4.2.4; when the value of this attribute is 'single-document' or 'single-document-new-
5626 sheet', the print-stream pages of document data from subsequent documents is used to complete
5627 the impression.

5628
5629 The size(scaling), position(translation) and rotation of the print-stream pages on the impression is
5630 implementation defined. Note that during this process the print-stream pages may be rendered to
5631 a form suitable for placing on the impression; this rendering is controlled by the values of the
5632 "printer-resolution" and "print-quality" attributes as described in sections 4.2.12 and 4.2.13. In
5633 the case N=1, the impression is nearly the same as the print-stream page; the differences would
5634 only be in the size, position and rotation of the print-stream page and/or any decoration, such as a
5635 frame to the page, that is added by the implementation.

5636
5637 4. The collection of impressions is placed, in sequence, onto sides of the media sheets. This
5638 placement is controlled by the "sides" attribute and the orientation of the print-stream page, as
5639 described in section 4.2.8. The orientation of the print-stream pages affects the orientation of the
5640 impression; for example, if "number-up" equals 2, then, typically, two portrait print-stream pages
5641 become one landscape impression. Note that the placement of impressions onto media sheets is
5642 also controlled by the "multiple-document-handling" attribute as described in section 4.2.4.

5643
5644 5. The "copies" and "multiple-document-handling" attributes are used to determine how many copies
5645 of each media instance are created and in what order. See sections 4.2.5 and 4.2.4 for the details.

5646
5647 6. When the correct number of copies are created, the media instances are finished according to the
5648 values of the "finishings" attribute as described in 4.2.6. Note that sometimes finishing
5649 operations may require manual intervention to perform the finishing operations on the copies,
5650 especially uncollated copies. This specification allows any or all of the processing steps to be
5651 performed automatically or manually at the discretion of the Printer object.

5652 17. APPENDIX E: Generic Directory Schema

5653 This section defines a generic schema for an entry in a directory service. A directory service is a means
5654 by which service users can locate service providers. In IPP environments, this means that IPP Printers
5655 can be registered (either automatically or with the help of an administrator) as entries of type printer in
5656 the directory using an implementation specific mechanism such as entry attributes, entry type fields,
5657 specific branches, etc. IPP clients can search or browse for entries of type printer. Clients use the
5658 directory service to find entries based on naming, organizational contexts, or filtered searches on
5659 attribute values of entries. For example, a client can find all printers in the "Local Department" context.
5660 Authentication and authorization are also often part of a directory service so that an administrator can

5661 place limits on end users so that they are only allowed to find entries to which they have certain access
5662 rights. IPP itself does not require any specific directory service protocol or provider.

5663 Note: Some directory implementations allow for the notion of "aliasing". That is, one directory entry
5664 object can appear as multiple directory entry object with different names for each object. In each case,
5665 each alias refers to the same directory entry object which refers to a single IPP Printer object.

5666 The generic schema is a subset of IPP Printer Job Template and Printer Description attributes (sections
5667 4.2 and 4.4). These attributes are identified as either RECOMMENDED or OPTIONAL for the
5668 directory entry itself. This conformance labeling is NOT the same conformance labeling applied to the
5669 attributes of IPP Printers objects. The conformance labeling in this Appendix is intended to apply to
5670 directory templates and to IPP Printer implementations that subscribe by adding one or more entries to a
5671 directory. RECOMMENDED attributes SHOULD be associated with each directory entry. OPTIONAL
5672 attributes MAY be associated with the directory entry (if known or supported). In addition, all directory
5673 entry attributes SHOULD reflect the current attribute values for the corresponding Printer object.

5674 The names of attributes in directory schema and entries SHOULD be the same as the IPP Printer
5675 attribute names as shown.

5676 In order to bridge between the directory service and the IPP Printer object, one of the RECOMMENDED
5677 directory entry attributes is the Printer object's "printer-uri-supported" attribute. The IPP client queries
5678 the "printer-uri-supported" attribute in the directory entry and then addresses the IPP Printer object using
5679 one of its URIs. The "uri-security-supported" attribute identifies the protocol (if any) used to secure a
5680 channel.

5681 The following attributes define the generic schema for directory entries of type PRINTER:

5682	printer-uri-supported	RECOMMENDED	Section 4.4.1
5683	uri-security-supported	RECOMMENDED	Section 4.4.2
5684	printer-name	RECOMMENDED	Section 4.4.3
5685	printer-location	RECOMMENDED	Section 4.4.4
5686	printer-info	OPTIONAL	Section 4.4.5
5687	printer-more-info	OPTIONAL	Section 4.4.6
5688	printer-make-and-model	RECOMMENDED	Section 4.4.8
5689	charset-supported	OPTIONAL	Section 4.4.15
5690	generated-natural-language-		
5691	supported	OPTIONAL	Section 4.4.17
5692	document-format-supported	RECOMMENDED	Section 4.4.19
5693	<u>compression-supported</u>	<u>RECOMMENDED</u>	<u>Section 4.4.29</u>
5694	color-supported	RECOMMENDED	Section 4.4.23
5695	finishings-supported	OPTIONAL	Section 4.2.6
5696	number-up-supported	OPTIONAL	Section 4.2.7
5697	sides-supported	RECOMMENDED	Section 4.2.8
5698	media-supported	RECOMMENDED	Section 4.2.11
5699	printer-resolution-supported	OPTIONAL	Section 4.2.12
5700	print-quality-supported	OPTIONAL	Section 4.2.13
5701	pages-per-minute	OPTIONAL	Section 4.4.33

5702 pages-per-minute-color OPTIONAL Section 4.4.34

5703

5704 18. APPENDIX F: Differences between the IPP/1.0 and IPP/1.1 "Model and Semantics" Specifications

5705 The following IPP/1.0 [IPP-MOD1.0] extensions and clarifications have been incorporated into IPP/1.1:

- 5706 1. Section 3.1.7 - clarified that only the version number parameter will be carried forward into
5707 future major or minor versions of the protocol.
- 5708 2. Section 3.2.1.1 - clarified that the Printer object rejects a Print-Job request if it does not support
5709 the "compression" operation attribute and a client supplies it.
- 5710 3. Sections 3.2.7, 3.2.8, and 3.2.9 - added the OPTIONAL Pause-Printer, Resume-Printer, and
5711 Purge-Jobs operations
- 5712 4. Sections 3.3.5, 3.3.6, and 3.3.7 - added the OPTIONAL Hold-Job, Release-Job, and Restart-Job
5713 operations.
- 5714 5. Section 4.1.9 - added 'image-tiff' and 'application/pdf' values.
- 5715 6. Section 4.2.2 - added the 'indefinite' keyword value to the "job-hold-until" attribute for use with
5716 the create operations and Hold-Job and Restart-Job operations.
- 5717 7. Section 4.2.6 - added more enum values to the "finishings" Job Template attribute.
- 5718 8. Section 4.3.7.1 - added the Partitioning of Job States section.
- 5719 9. Section 4.3.8 - added the 'job-restartable' keyword value to the "job-state-reasons" attribute for
5720 use with the Restart-Job operation.
- 5721 10. Section 4.4.2 - added the 'tls' keyword value to the "uri-security-supported" attribute.
- 5722 11. Section 4.4.11 - added the 'moving-to-paused' keyword value to the "printer-state-reasons"
5723 attribute for use with the Pause-Job operation.
- 5724 12. Section 4.4.11 - replaced the duplicate 'marker-supply-low' keyword with the missing 'toner-
5725 empty' keyword for the "printer-state-reasons" attribute.
- 5726 13. Section 4.4.13 - added the enum values to the "operations-supported" attribute for the new
5727 operations. Clarified that the values of this attribute are encoded as any enum, namely 32-bit
5728 values.
- 5729 14. Sections 4.4.33 and 4.4.34 - added the OPTIONAL "pages-per-minute" and "pages-per-minute-
5730 color" Printer Description attributes.
- 5731 15. Section 8.5 - added the security discussion around the new operator operations.
- 5732 16. Section 17 - added the OPTIONAL "pages-per-minute" and "pages-per-minute-color" Printer
5733 attributes to the Directory schema.

5734 The following changes were made to IPP/1.0 [IPP-MOD1.0] to create this IPP/1.1 document:

- 5735 1. Section 3.1.7, 5.2.4, and 14.1.5.4 - IPP objects MUST support both version 1.0 and 1.1. Clients
5736 MUST support version 1.1 and MAY support version 1.0.
- 5737 2. Section 4.1.9 - deleted 'text/plain; charset=iso-10646-ucs-2', since binary is not legal with the
5738 'text' type.
- 5739 3. Section 5.4, 8.2, and 8.7 - changed the IPP object security requirements from OPTIONAL non-
5740 standards track SSL3 to RECOMMENDED standards track TLS. Changed the client security

5741 requirements from RECOMMENDED non-standards track SSL3 to RECOMMENDED
5742 standards track TLS

5743 See also the "IPP/1.1 Encoding and Transport" [ipp-pro] document for differences between IPP/1.0 [IPP-
5744 PRO1.0] and IPP/1.1 [IPP-PRO].

5745
5746 INTERNET-DRAFT IPP/1.1: Model and Semantics April 6, 1999

5747 deBry, Hastings, Herriot, Isaacson, Powell [Page {PAGE|150}]
5748 Expires October 6, 1999

5750
5751 deBry, Hastings, Herriot, Isaacson, Powell [Page {PAGE|1}]
5752 Expires October 6, 1999

5753
5754
5755