

1 INTERNET-DRAFT  
2 draft-ietf-ipp-model-v11-01.txt

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

3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15 Internet Printing Protocol/1.1: Model and Semantics  
16 Copyright (C) The Internet Society (1999). All Rights Reserved.

17 Status of this Memo

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

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

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

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

27 Abstract

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

37 The full set of IPP documents includes:

38 Design Goals for an Internet Printing Protocol [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 .....	31
89	3.2.1.1 Print-Job Request .....	31
90	3.2.1.2 Print-Job Response .....	35
91	3.2.2 Print-URI Operation .....	37
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 .....	43
100	3.2.7 Pause-Printer Operation .....	44
101	3.2.7.1 Pause-Printer Request .....	45
102	3.2.7.2 Pause-Printer Response .....	46
103	3.2.8 Resume-Printer Operation .....	46
104	3.2.9 Purge-Jobs Operation .....	47
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 .....	50

109	3.3.2	Send-URI Operation .....	51
110	3.3.3	Cancel-Job Operation .....	51
111	3.3.3.1	Cancel-Job Request .....	51
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 .....	53
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 .....	55
118	3.3.5.2	Hold-Job Response .....	56
119	3.3.6	Release-Job Operation .....	56
120	3.3.7	Restart-Job Operation .....	57
121	3.3.7.1	Restart-Job Request .....	58
122	3.3.7.2	Restart-Job Response .....	59
123	4.	Object Attributes .....	59
124	4.1	Attribute Syntaxes .....	60
125	4.1.1	'text' 60	
126	4.1.1.1	'textWithoutLanguage' .....	61
127	4.1.1.2	'textWithLanguage' .....	61
128	4.1.2	'name' 62	
129	4.1.2.1	'nameWithoutLanguage' .....	62
130	4.1.2.2	'nameWithLanguage' .....	62
131	4.1.2.3	Matching 'name' attribute values .....	63
132	4.1.3	'keyword' .....	63
133	4.1.4	'enum' 64	
134	4.1.5	'uri' 65	
135	4.1.6	'uriScheme' .....	65
136	4.1.7	'charset' .....	65
137	4.1.8	'naturalLanguage' .....	66
138	4.1.9	'mimeType' .....	66
139	4.1.10	'octetString' .....	67
140	4.1.11	'boolean' .....	67
141	4.1.12	'integer' .....	68
142	4.1.13	'rangeOfInteger' .....	68
143	4.1.14	'dateTime' .....	68
144	4.1.15	'resolution' .....	68
145	4.1.16	'1setOf X' .....	68
146	4.2	Job Template Attributes .....	69
147	4.2.1	job-priority (integer(1:100)) .....	72
148	4.2.2	job-hold-until (type3 keyword   name (MAX)) .....	73
149	4.2.3	job-sheets (type3 keyword   name(MAX)) .....	73
150	4.2.4	multiple-document-handling (type2 keyword) .....	74
151	4.2.5	copies (integer(1:MAX)) .....	75
152	4.2.6	finishings (1setOf type2 enum) .....	75

153	4.2.7	page-ranges (1setOf rangeOfInteger (1:MAX)) .....	77
154	4.2.8	sides (type2 keyword).....	78
155	4.2.9	number-up (integer(1:MAX)).....	78
156	4.2.10	orientation-requested (type2 enum).....	79
157	4.2.11	media (type3 keyword   name(MAX)).....	80
158	4.2.12	printer-resolution (resolution) .....	80
159	4.2.13	print-quality (type2 enum).....	80
160	4.3	Job Description Attributes .....	81
161	4.3.1	job-uri (uri) .....	83
162	4.3.2	job-id (integer(1:MAX)).....	83
163	4.3.3	job-printer-uri (uri) .....	83
164	4.3.4	job-more-info (uri).....	83
165	4.3.5	job-name (name(MAX)).....	84
166	4.3.6	job-originating-user-name (name(MAX)).....	84
167	4.3.7	job-state (type1 enum).....	84
168	4.3.7.1	Partitioning of Job States .....	87
169	4.3.8	job-state-reasons (1setOf type2 keyword) .....	87
170	4.3.9	job-state-message (text(MAX)).....	89
171	4.3.10	number-of-documents (integer(0:MAX)).....	90
172	4.3.11	output-device-assigned (name(127)) .....	90
173	4.3.12	time-at-creation (integer(0:MAX)).....	90
174	4.3.13	time-at-processing (integer(0:MAX)).....	90
175	4.3.14	time-at-completed (integer(0:MAX)) .....	90
176	4.3.15	number-of-intervening-jobs (integer(0:MAX)) .....	90
177	4.3.16	job-message-from-operator (text(127)) .....	90
178	4.3.17	job-k-octets (integer(0:MAX)) .....	91
179	4.3.18	job-impressions (integer(0:MAX)).....	91
180	4.3.19	job-media-sheets (integer(0:MAX)) .....	92
181	4.3.20	job-k-octets-processed (integer(0:MAX)) .....	92
182	4.3.21	job-impressions-completed (integer(0:MAX)).....	92
183	4.3.22	job-media-sheets-completed (integer(0:MAX)) .....	92
184	4.3.23	attributes-charset (charset).....	93
185	4.3.24	attributes-natural-language (naturalLanguage).....	93
186	4.4	Printer Description Attributes.....	93
187	4.4.1	printer-uri-supported (1setOf uri).....	95
188	4.4.2	uri-security-supported (1setOf type2 keyword).....	95
189	4.4.3	printer-name (name(127)).....	96
190	4.4.4	printer-location (text(127)) .....	97
191	4.4.5	printer-info (text(127)) .....	97
192	4.4.6	printer-more-info (uri) .....	97
193	4.4.7	printer-driver-installer (uri) .....	97
194	4.4.8	printer-make-and-model (text(127)).....	97
195	4.4.9	printer-more-info-manufacturer (uri) .....	97
196	4.4.10	printer-state (type1 enum) .....	98
197	4.4.11	printer-state-reasons (1setOf type2 keyword).....	99

198	4.4.12	printer-state-message (text(MAX)) .....	101
199	4.4.13	operations-supported (1setOf type2 enum) .....	101
200	4.4.14	charset-configured (charset) .....	102
201	4.4.15	charset-supported (1setOf charset) .....	102
202	4.4.16	natural-language-configured (naturalLanguage) .....	102
203	4.4.17	generated-natural-language-supported (1setOf naturalLanguage).....	102
204	4.4.18	document-format-default (mimeMediaType).....	103
205	4.4.19	document-format-supported (1setOf mimeMediaType).....	103
206	4.4.20	printer-is-accepting-jobs (boolean).....	103
207	4.4.21	queued-job-count (integer(0:MAX)) .....	103
208	4.4.22	printer-message-from-operator (text(127)).....	103
209	4.4.23	color-supported (boolean) .....	104
210	4.4.24	reference-uri-schemes-supported (1setOf uriScheme) .....	104
211	4.4.25	pdl-override-supported (type2 keyword).....	104
212	4.4.26	printer-up-time (integer(1:MAX)).....	104
213	4.4.27	printer-current-time (dateTime).....	105
214	4.4.28	multiple-operation-time-out (integer(1:MAX)).....	105
215	4.4.29	compression-supported (1setOf type3 keyword).....	105
216	4.4.30	job-k-octets-supported (rangeOfInteger(0:MAX)) .....	105
217	4.4.31	job-impressions-supported (rangeOfInteger(0:MAX)).....	106
218	4.4.32	job-media-sheets-supported (rangeOfInteger(0:MAX)).....	106
219	4.4.33	pages-per-minute (integer(0:MAX)) .....	106
220	4.4.34	pages-per-minute-color (integer(0:MAX)).....	106
221	5.	Conformance .....	107
222	5.1	Client Conformance Requirements.....	107
223	5.2	IPP Object Conformance Requirements .....	107
224	5.2.1	Objects.....	108
225	5.2.2	Operations.....	108
226	5.2.3	IPP Object Attributes.....	108
227	5.2.4	Versions.....	107
228	5.2.5	Extensions .....	109
229	5.2.6	Attribute Syntaxes .....	109
230	5.3	Charset and Natural Language Requirements.....	109
231	5.4	Security Conformance Requirements .....	110
232	6.	IANA Considerations (registered and private extensions).....	110
233	6.1	Typed 'keyword' and 'enum' Extensions .....	110
234	6.2	Attribute Extensibility .....	112
235	6.3	Attribute Syntax Extensibility .....	113
236	6.4	Operation Extensibility.....	113
237	6.5	Attribute Groups .....	113
238	6.6	Status Code Extensibility.....	114
239	6.7	Registration of MIME types/sub-types for document-formats .....	114
240	6.8	Registration of charsets for use in 'charset' attribute values .....	114

241	7. Internationalization Considerations.....	115
242	8. Security Considerations.....	117
243	8.1 Security Scenarios.....	118
244	8.1.1 Client and Server in the Same Security Domain.....	118
245	8.1.2 Client and Server in Different Security Domains.....	118
246	8.1.3 Print by Reference.....	119
247	8.2 URIs for TLS and non-TLS Access.....	119
248	8.3 The "requesting-user-name" (name(MAX)) Operation Attribute.....	119
249	8.4 Restricted Queries.....	121
250	8.5 Operations performed by operators and system administrators.....	121
251	8.6 Queries on jobs submitted using non-IPP protocols.....	121
252	8.7 IPP Security Application Profile for TLS.....	121
253	9. References.....	122
254	10. Notices.....	125
255	11. Author's Address.....	126
256	12. Formats for IPP Registration Proposals.....	129
257	12.1 Type2 keyword attribute values registration.....	129
258	12.2 Type3 keyword attribute values registration.....	129
259	12.3 Type2 enum attribute values registration.....	129
260	12.4 Type3 enum attribute values registration.....	130
261	12.5 Attribute registration.....	130
262	12.6 Attribute Syntax registration.....	131
263	12.7 Operation registration.....	131
264	12.8 Attribute Group registration.....	131
265	12.9 Status code registration.....	132
266	13. APPENDIX A: Terminology.....	132
267	13.1 Conformance Terminology.....	132
268	13.1.1 NEED NOT.....	132
269	13.2 Model Terminology.....	132
270	13.2.1 Keyword.....	132
271	13.2.2 Attributes.....	132
272	13.2.2.1 Attribute Name.....	133
273	13.2.2.2 Attribute Group Name.....	133
274	13.2.2.3 Attribute Value.....	133
275	13.2.2.4 Attribute Syntax.....	133
276	13.2.3 Supports.....	133
277	13.2.4 print-stream page.....	135
278	13.2.5 impression.....	135
279	14. APPENDIX B: Status Codes and Suggested Status Code Messages.....	135

280	14.1	Status Codes .....	136
281	14.1.1	Informational .....	136
282	14.1.2	Successful Status Codes .....	136
283	14.1.2.1	successful-ok (0x0000).....	136
284	14.1.2.2	successful-ok-ignored-or-substituted-attributes (0x0001).....	137
285	14.1.2.3	successful-ok-conflicting-attributes (0x0002).....	137
286	14.1.3	Redirection Status Codes.....	137
287	14.1.4	Client Error Status Codes .....	137
288	14.1.4.1	client-error-bad-request (0x0400).....	137
289	14.1.4.2	client-error-forbidden (0x0401).....	138
290	14.1.4.3	client-error-not-authenticated (0x0402).....	138
291	14.1.4.4	client-error-not-authorized (0x0403).....	138
292	14.1.4.5	client-error-not-possible (0x0404).....	138
293	14.1.4.6	client-error-timeout (0x0405).....	138
294	14.1.4.7	client-error-not-found (0x0406).....	138
295	14.1.4.8	client-error-gone (0x0407).....	139
296	14.1.4.9	client-error-request-entity-too-large (0x0408).....	139
297	14.1.4.10	client-error-request-value-too-long (0x0409).....	139
298	14.1.4.11	client-error-document-format-not-supported (0x040A).....	140
299	14.1.4.12	client-error-attributes-or-values-not-supported (0x040B).....	140
300	14.1.4.13	client-error-uri-scheme-not-supported (0x040C).....	140
301	14.1.4.14	client-error-charset-not-supported (0x040D).....	140
302	14.1.4.15	client-error-conflicting-attributes (0x040E).....	140
303	14.1.5	Server Error Status Codes .....	140
304	14.1.5.1	server-error-internal-error (0x0500).....	141
305	14.1.5.2	server-error-operation-not-supported (0x0501).....	141
306	14.1.5.3	server-error-service-unavailable (0x0502).....	141
307	14.1.5.4	server-error-version-not-supported (0x0503).....	141
308	14.1.5.5	server-error-device-error (0x0504).....	141
309	14.1.5.6	server-error-temporary-error (0x0505).....	142
310	14.1.5.7	server-error-not-accepting-jobs (0x0506).....	142
311	14.1.5.8	server-error-busy (0x0507).....	142
312	14.1.5.9	server-error-job-canceled (0x0508).....	142
313	14.2	Status Codes for IPP Operations.....	143
314	15.	APPENDIX C: "media" keyword values .....	144
315	16.	APPENDIX D: Processing IPP Attributes .....	148
316	16.1	Fidelity.....	149
317	16.2	Page Description Language (PDL) Override .....	150
318	16.3	Using Job Template Attributes During Document Processing.....	151
319	17.	APPENDIX E: Generic Directory Schema .....	152
320	18.	APPENDIX F: Differences between the IPP/1.0 and IPP/1.1 "Model and Semantics" Specifications .....	154
321			

322

323

## 324 1. Introduction

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

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

335

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

338 This document is laid out as follows:

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

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

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

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

## 365 1.1 Simplified Printing Model

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

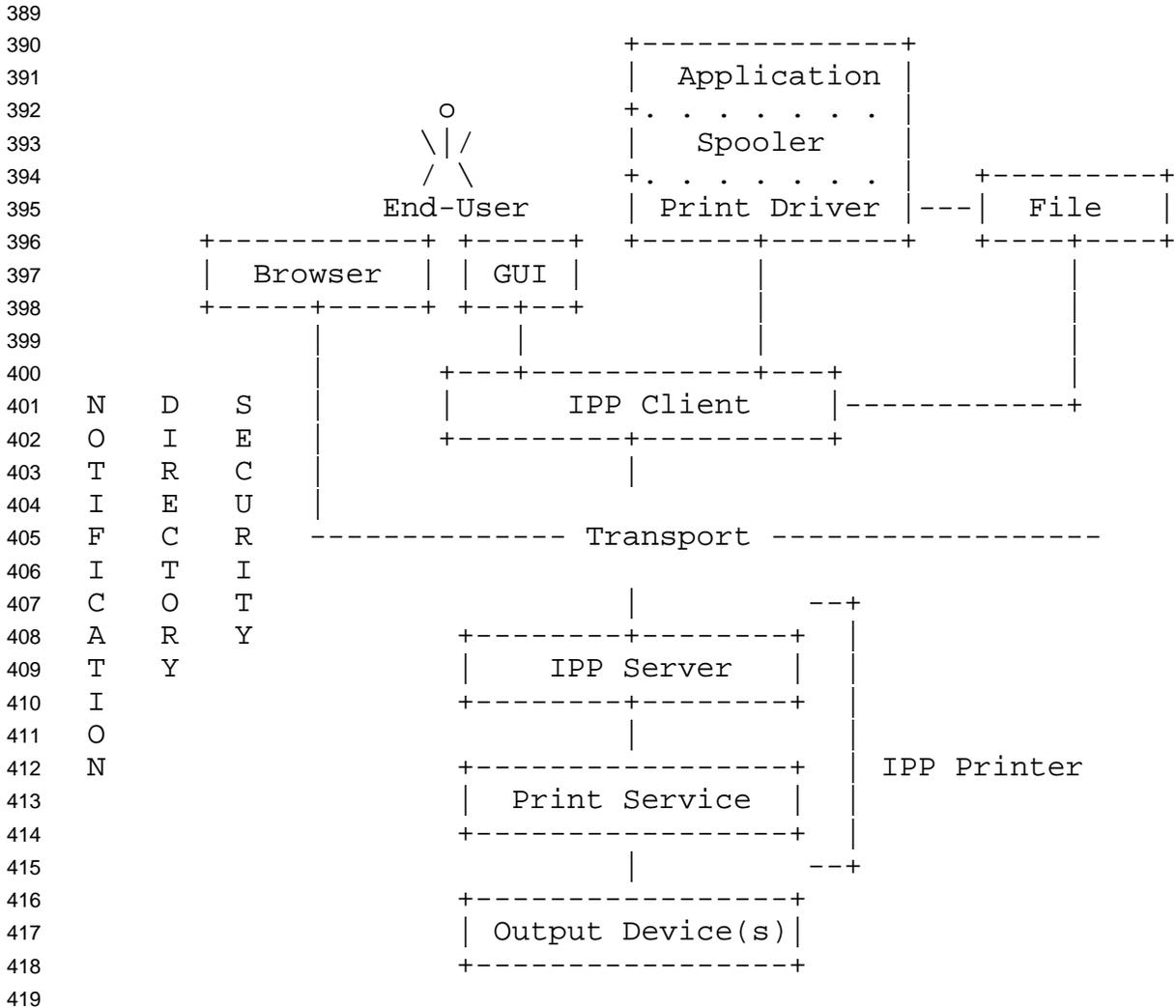
380 The IPP/1.1 model encapsulates the important components of distributed printing into two object types:

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

383

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

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



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

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

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

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

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

## 450 2. IPP Objects

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

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

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

462

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

### 465 2.1 Printer Object

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

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

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

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

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

482 Some examples of configurations supporting a Printer object include:

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

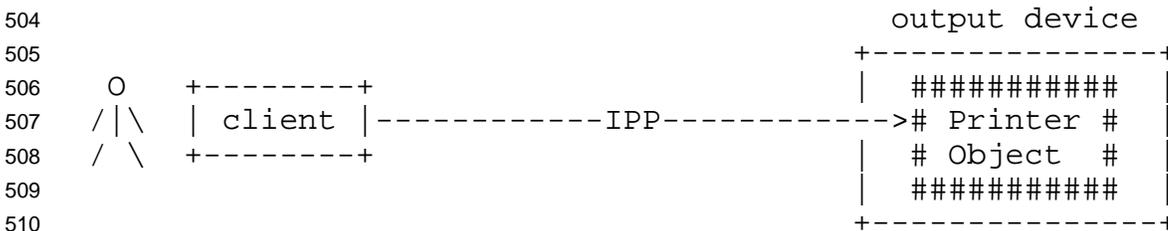
489 The following figures show some examples of how Printer objects can be realized on top of various  
490 distributed printing configurations. The embedded case below represents configurations 1 and 2. The  
491 hosted and fan-out figures below represent configurations 3a and 3b.

492 Legend:

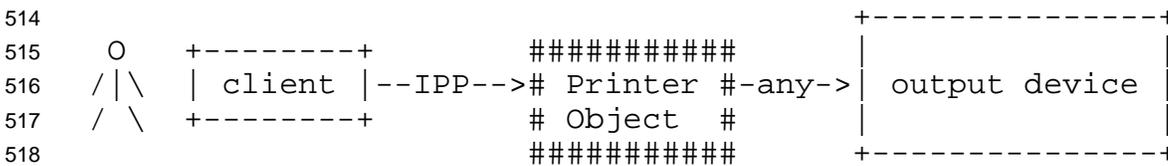
493  
494 ##### indicates a Printer object which is  
495 either embedded in an output device or is  
496 hosted in a server. The Printer object  
497 might or might not be capable of queuing/spooling.

498  
499 any indicates any network protocol or direct  
500 connect, including IPP

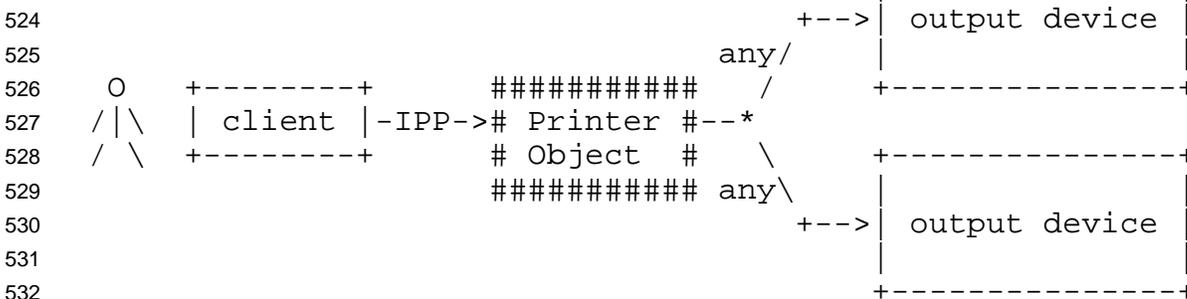
501  
502  
503 embedded printer:



511  
512  
513 hosted printer:



519  
520  
521  
522  
523 fan out:



535 2.2 Job Object

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

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

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

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

548  
549 An implementation **MUST** support at least one document per Job object. An implementation **MAY**  
550 support multiple documents per Job object. A document is either:

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

554

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

## 558 2.3 Object Relationships

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

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

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

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

## 574 2.4 Object Identity

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

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

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

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

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

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

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

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

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

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

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

651 To summarize:

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

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

### 672 3. IPP Operations

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

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

686 The IPP/1.1 Printer operations are:

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

694 Resume-Printer (section 3.3.6)

695 Purge-Jobs (section 3.3.7)

696

697 The Job operations are:

698 Send-Document (section 3.3.1)

699 Send-URI (section 3.3.2)

700 Cancel-Job (section 3.3.3)

701 Get-Job-Attributes (section 3.3.4)

702 Hold-Job (section 3.3.5)

703 Release-Job (section 3.3.6)

704 Restart-Job (section 3.3.7)

705

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

### 708 3.1 Common Semantics

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

#### 711 3.1.1 Required Parameters

712 Every operation request contains the following REQUIRED parameters:

713 - a "version-number",

714 - an "operation-id",

715 - a "request-id", and

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

717

718 Every operation response contains the following REQUIRED parameters:

719 - a "version-number",

720 - a "status-code",

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

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

723

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

#### 727 3.1.2 Operation IDs and Request IDs

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

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

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

### 743 3.1.3 Attributes

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

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

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

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

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

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

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

#### 797   3.1.4 Character Set and Natural Language Operation Attributes

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

## 811 3.1.4.1 Request Operation Attributes

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

814 "attributes-charset" (charset):

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

820  
821 All clients and IPP objects **MUST** support the 'utf-8' charset [RFC2279] and **MAY** support  
822 additional charsets provided that they are registered with IANA [IANA-CS]. If the Printer object  
823 does not support the client supplied charset value, the Printer object **MUST** reject the request, set  
824 the "attributes-charset" to 'utf-8' in the response, and return the 'client-error-charset-not-  
825 supported' status code and any 'text' or 'name' attributes using the 'utf-8' charset. The Printer  
826 object **MUST** indicate the charset(s) supported as the values of the "charset-supported" Printer  
827 attribute (see Section 4.4.15), so that the client can query to determine which charset(s) are  
828 supported.

829  
830 Note to client implementers: Since IPP objects are only required to support the 'utf-8' charset, in  
831 order to maximize interoperability with multiple IPP object implementations, a client may want  
832 to supply 'utf-8' in the "attributes-charset" operation attribute, even though the client is only  
833 passing and able to present a simpler charset, such as US-ASCII or ISO-8859-1. Then the client  
834 will have to filter out (or charset convert) those characters that are returned in the response that it  
835 cannot present to its user. On the other hand, if both the client and the IPP objects also support a  
836 charset in common besides utf-8, the client may want to use that charset in order to avoid charset  
837 conversion or data loss.

838  
839 See the 'charset' attribute syntax description in Section 4.1.7 for the syntax and semantic  
840 interpretation of the values of this attribute and for example values.

841  
842 "attributes-natural-language" (naturalLanguage):

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

847  
848 There are no **REQUIRED** natural languages required for the Printer object to support. However,  
849 the Printer object's "generated-natural-language-supported" attribute identifies the natural  
850 languages supported by the Printer object and any contained Job objects for all text strings  
851 generated by the IPP object. A client **MAY** query this attribute to determine which natural  
852 language(s) are supported for generated messages.

853

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

860  
861 For other 'text' and 'name' attributes supplied by the client, authentication system, operator,  
862 system administrator, or manufacturer (i.e., for "job-originating-user-name", "printer-name"  
863 (name), "printer-location" (text), "printer-info" (text), and "printer-make-and-model" (text)), the  
864 Printer object is only required to support the configured natural language of the Printer identified  
865 by the Printer object's "natural-language-configured" attribute, though support of additional  
866 natural languages for these attributes is permitted.

867  
868 For any 'text' or 'name' attribute in the request that is in a different natural language than the value  
869 supplied in the "attributes-natural-language" operation attribute, the client MUST use the Natural  
870 Language Override mechanism (see sections 4.1.1.2 and 4.1.2.2) for each such attribute value  
871 supplied. The client MAY use the Natural Language Override mechanism redundantly, i.e., use  
872 it even when the value is in the same natural language as the value supplied in the "attributes-  
873 natural-language" operation attribute of the request.

874  
875 The IPP object MUST accept any natural language and any Natural Language Override, whether  
876 the IPP object supports that natural language or not (and independent of the value of the "ipp-  
877 attribute-fidelity" Operation attribute). That is the IPP object accepts all client supplied values no  
878 matter what the values are in the Printer object's "generated-natural-language-supported"  
879 attribute. That attribute, "generated-natural-language-supported", only applies to generated  
880 messages, not client supplied messages. The IPP object MUST remember that natural language  
881 for all client-supplied attributes, and when returning those attributes in response to a query, the  
882 IPP object MUST indicate that natural language.

883  
884 Each value whose attribute syntax type is 'text' or 'name' (see sections 4.1.1 and 4.1.2) has an  
885 Associated Natural-Language. This document does not specify how this association is stored in a  
886 Printer or Job object. When such a value is encoded in a request or response, the natural  
887 language is either implicit or explicit:

- 888  
889 • In the implicit case, the value contains only the text/name value, and the language is  
890 specified by the "attributes-natural-language" operation attribute in the request or  
891 response (see sections 4.1.1.1 textWithoutLanguage and 4.1.2.1  
892 nameWithoutLanguage).
- 893  
894 • In the explicit case (also known as the Natural-Language Override case), the value  
895 contains both the language and the text/name value (see sections 4.1.1.2  
896 textWithLanguage and 4.1.2.2 nameWithLanguage).

897

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

908  
909 An IPP object MUST NOT reject a request based on a supplied natural language in an  
910 "attributes-natural-language" Operation attribute or in any attribute that uses the Natural  
911 Language Override.

912  
913 See the 'naturalLanguage' attribute syntax description in section 4.1.8 for the syntax and semantic  
914 interpretation of the values of this attribute and for example values.

915  
916 Clients SHOULD NOT supply 'text' or 'name' attributes that use an illegal combination of natural  
917 language and charset. For example, suppose a Printer object supports charsets 'utf-8', 'iso-8859-1', and  
918 'iso-8859-7'. Suppose also, that it supports natural languages 'en' (English), 'fr' (French), and 'el' (Greek).  
919 Although the Printer object supports the charset 'iso-8859-1' and natural language 'el', it probably does  
920 not support the combination of Greek text strings using the 'iso-8859-1' charset. The Printer object  
921 handles this apparent incompatibility differently depending on the context in which it occurs:

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

930  
931 In either case, the Printer object does not reject the request because of the apparent incompatibility. The  
932 potential incompatible combination of charset and natural language can occur either at the global  
933 operation level or at the Natural Language Override attribute-by-attribute level. In addition, since the  
934 response always includes explicit charset and natural language information, there is never any question  
935 or ambiguity in how the client interprets the response.

#### 936 3.1.4.2 Response Operation Attributes

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

939 "attributes-charset" (charset):

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

945  
946 If the Printer object supports more than just the 'utf-8' charset, the Printer object **MUST** be able to  
947 code convert between each of the charsets supported on a highest fidelity possible basis in order  
948 to return the 'text' and 'name' attributes in the charset requested by the client. However, some  
949 information loss **MAY** occur during the charset conversion depending on the charsets involved.  
950 For example, the Printer object may convert from a UTF-8 'a' to a US-ASCII 'a' (with no loss of  
951 information), from an ISO Latin 1 CAPITAL LETTER A WITH ACUTE ACCENT to US-  
952 ASCII 'A' (losing the accent), or from a UTF-8 Japanese Kanji character to some ISO Latin 1  
953 error character indication such as '?', decimal code equivalent, or to the absence of a character,  
954 depending on implementation.

955  
956 Note: Whether an implementation that supports more than one charset stores the data in the  
957 charset supplied by the client or code converts to one of the other supported charsets, depends on  
958 implementation. The strategy should try to minimize loss of information during code conversion.  
959 On each response, such an implementation converts from its internal charset to that requested.

960  
961 "attributes-natural-language" (naturalLanguage):

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

### 974 3.1.5 Operation Targets

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

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

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

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

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

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

1003 In all cases, the target URIs contained within the body of IPP operation requests and responses must be  
1004 in absolute format rather than relative format (a relative URL identifies a resource with the scope of the  
1005 HTTP server, but does not include scheme, host or port).

1006 The following rules apply to the use of port numbers in URIs that identify IPP objects:

- 1007 1. If the URI scheme allows the port number to be explicitly included in the URI string, and a port  
1008 number is specified within the URI, then that port number MUST be used by the client to contact  
1009 the IPP object.
- 1010
- 1011 2. If the URI scheme allows the port number to be explicitly included in the URI string, and a port  
1012 number is not specified within the URI, then default port number implied by that URI scheme  
1013 MUST be used by the client to contact the IPP object.
- 1014
- 1015 3. If the URI scheme does not allow an explicit port number to be specified within the URI, then the  
1016 default port number implied by that URI MUST be used by the client to contact the IPP object.
- 1017

1018 Note: The IPP "Encoding and Transport document [IPP-PRO] shows a mapping of IPP onto HTTP/1.1  
1019 and defines a new default port number for using IPP over HTTP/1.1.

## 1020 3.1.6 Operation Status Codes and Messages

1021 Every operation response includes a REQUIRED "status-code" parameter and an OPTIONAL "status-  
1022 message" operation attribute. The "status-code" provides information on the processing of a request. A  
1023 "status-message" attribute provides a short textual description of the status of the operation. The status  
1024 code is intended for use by automata, and the status message is intended for the human end user. If a  
1025 response does include a "status-message" attribute, an IPP client NEED NOT examine or display the  
1026 message, however it SHOULD do so in some implementation specific manner.

1027 The "status-code" value is a numeric value that has semantic meaning. The "status-code" syntax is  
1028 similar to a "type2 enum" (see section 4.1 on "Attribute Syntaxes") except that values can range only  
1029 from 0x0000 to 0x7FFF. Section 14 describes the status codes, assigns the numeric values, and suggests  
1030 a corresponding status message for each status code. The "status-message" attribute's syntax is  
1031 "text(255)". A client implementation of IPP SHOULD convert status code values into any localized  
1032 message that has semantic meaning to the end user.

1033 If the Printer object supports the "status-message" operation attribute, the Printer object MUST be able  
1034 to generate this message in any of the natural languages identified by the Printer object's "generated-  
1035 natural-language-supported" attribute (see the "attributes-natural-language" operation attribute specified  
1036 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  
1037 generating this message, the Printer object uses the natural language indicated by the value of the  
1038 "attributes-natural-language" in the client request if supported, otherwise the Printer object uses the  
1039 value in the Printer object's own "natural-language-configured" attribute. If the Printer object supports  
1040 the "status-message" operation attribute, it SHOULD use the REQUIRED 'utf-8' charset to return a status  
1041 message for the following error status codes (see section 14): 'client-error-bad-request', 'client-error-  
1042 charset-not-supported', 'server-error-internal-error', 'server-error-operation-not-supported', and 'server-  
1043 error-version-not-supported'. In this case, it MUST set the value of the "attributes-charset" operation  
1044 attribute to 'utf-8' in the error response.

## 1045 3.1.7 Versions

1046 Each operation request and response carries with it a "version-number" parameter. Each value of the  
1047 "version-number" is in the form "X.Y" where X is the major version number and Y is the minor version  
1048 number. By including a version number in the client request, it allows the client to identify which  
1049 version of IPP it is interested in using. If the IPP object does not support that version, the object  
1050 responds with a status code of 'server-error-version-not-supported' along with the closest version number  
1051 that is supported (see section 14.1.5.4).

1052 There is no version negotiation per se. However, if after receiving a 'server-error-version-not-supported'  
1053 status code from an IPP object, there is nothing that prevents a client from trying again with a different  
1054 version number. In order to conform to IPP/1.1, an IPP object implementations MUST support versions  
1055 '1.1' and 1.0.

1056 There is only one notion of "version number" that covers both IPP Model and IPP Protocol changes.  
1057 Thus the version number MUST change when introducing a new version of the Model and Semantics  
1058 document [IPP-MOD] or a new version of the "Encoding and Transport" document [IPP-PRO].

1059 Changes to the major version number indicate structural or syntactic changes that make it impossible for  
1060 older version of IPP clients and Printer objects to correctly parse and process the new or changed  
1061 attributes, operations and responses. If the major version number changes, the minor version numbers is  
1062 set to zero. As an example, adding the "ipp-attribute-fidelity" attribute (if it had not been part of version  
1063 '1.1'), would have required a change to the major version number. Items that might affect the changing  
1064 of the major version number include any changes to the Model and Semantics document [IPP-MOD] or  
1065 the "Encoding and Transport" document [IPP-PRO] itself, such as:

- 1066 - reordering of ordered attributes or attribute sets
- 1067 - changes to the syntax of existing attributes
- 1068 - changing Operation or Job Template attributes from OPTIONAL to REQUIRED and vice versa
- 1069 - adding REQUIRED (for an IPP object to support) operation attributes
- 1070 - adding REQUIRED (for an IPP object to support) operation attribute groups
- 1071 - adding values to existing operation attributes
- 1072 - adding REQUIRED operations

1073

1074 Changes to the minor version number indicate the addition of new features, attributes and attribute  
1075 values that may not be understood by all IPP objects, but which can be ignored if not understood. Items  
1076 that might affect the changing of the minor version number include any changes to the model objects and  
1077 attributes but not the encoding and transport rules [IPP-PRO] (except adding attribute syntaxes).

1078 Examples of such changes are:

- 1079 - grouping all extensions not included in a previous version into a new version
- 1080 - adding new attribute values
- 1081 - adding new object attributes
- 1082 - adding OPTIONAL (for an IPP object to support) operation attributes (i.e., those attributes that an  
1083 IPP object can ignore without confusing clients)
- 1084 - adding OPTIONAL (for an IPP object to support) operation attribute groups (i.e., those attributes  
1085 that an IPP object can ignore without confusing clients)
- 1086 - adding new attribute syntaxes
- 1087 - adding OPTIONAL operations
- 1088 - changing Job Description attributes or Printer Description attributes from OPTIONAL to  
1089 REQUIRED or vice versa.

1090

1091 The encoding of the "version-number" MUST NOT change over any version number (either major or  
1092 minor). This rule guarantees that all future versions will be backwards compatible with all previous  
1093 versions (at least for checking the "version-number"). In addition, any protocol elements (attributes,  
1094 error codes, tags, etc.) that are not carried forward from one version to the next are deprecated so that  
1095 they can never be reused with new semantics.

1096 Implementations that support a certain major version NEED NOT support ALL previous versions. As  
1097 each new major version is defined (through the release of a new specification), that major version will  
1098 specify which previous major versions MUST be supported in compliant implementations.

## 1099 3.1.8 Job Creation Operations

1100 In order to "submit a print job" and create a new Job object, a client issues a create request. A create  
1101 request is any one of following three operation requests:

- 1102 - The Print-Job Request: A client that wants to submit a print job with only a single document uses  
1103 the Print-Job operation. The operation allows for the client to "push" the document data to the  
1104 Printer object by including the document data in the request itself.  
1105
- 1106 - The Print-URI Request: A client that wants to submit a print job with only a single document  
1107 (where the Printer object "pulls" the document data instead of the client "pushing" the data to the  
1108 Printer object) uses the Print-URI operation. In this case, the client includes in the request only a  
1109 URI reference to the document data (not the document data itself).  
1110
- 1111 - The Create-Job Request: A client that wants to submit a print job with multiple documents uses the  
1112 Create-Job operation. This operation is followed by an arbitrary number of Send-Document  
1113 and/or Send-URI operations (each creating another document for the newly create Job object).  
1114 The Send-Document operation includes the document data in the request (the client "pushes" the  
1115 document data to the printer), and the Send-URI operation includes only a URI reference to the  
1116 document data in the request (the Printer "pulls" the document data from the referenced location).  
1117 The last Send-Document or Send-URI request for a given Job object includes a "last-document"  
1118 operation attribute set to 'true' indicating that this is the last request.  
1119

1120 Throughout this model specification, the term "create request" is used to refer to any of these three  
1121 operation requests.

1122 A Create-Job operation followed by only one Send-Document operation is semantically equivalent to a  
1123 Print-Job operation, however, for performance reasons, the client SHOULD use the Print-Job operation  
1124 for all single document jobs. Also, Print-Job is a REQUIRED operation (all implementations MUST  
1125 support it) whereas Create-Job is an OPTIONAL operation, hence some implementations might not  
1126 support it.

1127 Job submission time is the point in time when a client issues a create request. The initial state of every  
1128 Job object is the 'pending' or 'pending-held' state. Later, the Printer object begins processing the print  
1129 job. At this point in time, the Job object's state moves to 'processing'. This is known as job processing  
1130 time. There are validation checks that must be done at job submission time and others that must be  
1131 performed at job processing time.

1132 At job submission time and at the time a Validate-Job operation is received, the Printer MUST do the  
1133 following:

- 1134 1. Process the client supplied attributes and either accept or reject the request
- 1135 2. Validate the syntax of and support for the scheme of any client supplied URI

1136

1137 At job submission time the Printer object MUST validate whether or not the supplied attributes, attribute  
1138 syntaxes, and values are supported by matching them with the Printer object's corresponding "xxx-

1139 supported" attributes. See section 3.2.1.2 for details. [IPP-IIG] presents suggested steps for an IPP  
1140 object to either accept or reject any request and additional steps for processing create requests.

1141 At job submission time the Printer object NEED NOT perform the validation checks reserved for job  
1142 processing time such as:

- 1143 1. Validating the document data
  - 1144 2. Validating the actual contents of any client supplied URI (resolve the reference and follow the link  
1145 to the document data)
- 1146

1147 At job submission time, these additional job processing time validation checks are essentially useless,  
1148 since they require actually parsing and interpreting the document data, are not guaranteed to be 100%  
1149 accurate, and MUST be done, yet again, at job processing time. Also, in the case of a URI, checking for  
1150 availability at job submission time does not guarantee availability at job processing time. In addition, at  
1151 job processing time, the Printer object might discover any of the following conditions that were not  
1152 detectable at job submission time:

- 1153 - runtime errors in the document data,
  - 1154 - nested document data that is in an unsupported format,
  - 1155 - the URI reference is no longer valid (i.e., the server hosting the document might be down), or
  - 1156 - any other job processing error
- 1157

1158 At job processing time, since the Printer object has already responded with a successful status code in  
1159 the response to the create request, if the Printer object detects an error, the Printer object is unable to  
1160 inform the end user of the error with an operation status code. In this case, the Printer, depending on the  
1161 error, can set the "job-state", "job-state-reasons", or "job-state-message" attributes to the appropriate  
1162 value(s) so that later queries can report the correct job status.

1163 Note: Asynchronous notification of events is outside the scope of this IPP/1.1 document.

## 1164 3.2 Printer Operations

1165 All Printer operations are directed at Printer objects. A client MUST always supply the "printer-uri"  
1166 operation attribute in order to identify the correct target of the operation.

### 1167 3.2.1 Print-Job Operation

1168 This REQUIRED operation allows a client to submit a print job with only one document and supply the  
1169 document data (rather than just a reference to the data). See Section 16 for the suggested steps for  
1170 processing create operations and their Operation and Job Template attributes.

#### 1171 3.2.1.1 Print-Job Request

1172 The following groups of attributes are supplied as part of the Print-Job Request:

## 1173 Group 1: Operation Attributes

## 1174 Natural Language and Character Set:

1175 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1176 3.1.4.1. The Printer object **MUST** copy these values to the corresponding Job Description  
1177 attributes described in sections 4.3.23 and 4.3.24.

1178

## 1179 Target:

1180 The "printer-uri" (uri) operation attribute which is the target for this operation as described in  
1181 section 3.1.5.

1182

## 1183 Requesting User Name:

1184 The "requesting-user-name" (name(MAX)) attribute **SHOULD** be supplied by the client as  
1185 described in section 8.3.

1186

## 1187 "job-name" (name(MAX)):

1188 The client **OPTIONALLY** supplies this attribute. The Printer object **MUST** support this  
1189 attribute. It contains the client supplied Job name. If this attribute is supplied by the client, its  
1190 value is used for the "job-name" attribute of the newly created Job object. The client **MAY**  
1191 automatically include any information that will help the end-user distinguish amongst his/her  
1192 jobs, such as the name of the application program along with information from the document,  
1193 such as the document name, document subject, or source file name. If this attribute is not  
1194 supplied by the client, the Printer generates a name to use in the "job-name" attribute of the  
1195 newly created Job object (see Section 4.3.5).

1196

## 1197 "ipp-attribute-fidelity" (boolean):

1198 The client **OPTIONALLY** supplies this attribute. The Printer object **MUST** support this  
1199 attribute. The value 'true' indicates that total fidelity to client supplied Job Template attributes  
1200 and values is required, else the Printer object **MUST** reject the Print-Job request. The value  
1201 'false' indicates that a reasonable attempt to print the Job object is acceptable and the Printer  
1202 object **MUST** accept the Print-job request. If not supplied, the Printer object assumes the value is  
1203 'false'. All Printer objects **MUST** support both types of job processing. See section 16 for a full  
1204 description of "ipp-attribute-fidelity" and its relationship to other attributes, especially the Printer  
1205 object's "pdl-override-supported" attribute.

1206

## 1207 "document-name" (name(MAX)):

1208 The client **OPTIONALLY** supplies this attribute. The Printer object **MUST** support this  
1209 attribute. It contains the client supplied document name. The document name **MAY** be  
1210 different than the Job name. Typically, the client software automatically supplies the document  
1211 name on behalf of the end user by using a file name or an application generated name. If this  
1212 attribute is supplied, its value can be used in a manner defined by each implementation.  
1213 Examples include: printed along with the Job (job start sheet, page adornments, etc.), used by  
1214 accounting or resource tracking management tools, or even stored along with the document as a  
1215 document level attribute. IPP/1.1 does not support the concept of document level attributes.

1216

1217 "document-format" (mimeType) :

1218 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1219 attribute. The value of this attribute identifies the format of the supplied document data. If the  
1220 client does not supply this attribute, the Printer object assumes that the document data is in the  
1221 format defined by the Printer object's "document-format-default" attribute. If the client supplies  
1222 this attribute, but the value is not supported by the Printer object, i.e., the value is not one of the  
1223 values of the Printer object's "document-format-supported" attribute, the Printer object MUST  
1224 reject the request and return the 'client-error-document-format-not-supported' status code.

1225

1226 "document-natural-language" (naturalLanguage):

1227 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports  
1228 this attribute. This attribute specifies the natural language of the document for those document-  
1229 formats that require a specification of the natural language in order to image the document  
1230 unambiguously. There are no particular values required for the Printer object to support.

1231

1232 "compression" (type3 keyword)

1233 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports  
1234 this attribute and the "compression-supported" attribute (see section 4.4.29). The client supplied  
1235 "compression" operation attribute identifies the compression algorithm used on the document  
1236 data. If the client omits this attribute, the Printer object MUST assume that the data is not  
1237 compressed. If the client supplies the attribute and the Printer object supports the attribute, the  
1238 Printer object uses the corresponding decompression algorithm on the document data. If the  
1239 client supplies this attribute, but the value is not supported by the Printer object, i.e., the value is  
1240 not one of the values of the Printer object's "compression-supported" attribute, the Printer object  
1241 MUST copy the attribute and its value to the Unsupported Attributes response group, reject the  
1242 request, and return the 'client-error-attributes-or-values-not-supported' status code. If the client  
1243 supplies this attribute, but this attribute is not supported by the Printer object, i.e., the  
1244 "compression-supported" attribute is not one of the Printer's Printer Description attributes, the  
1245 Printer object MUST copy the attribute to the Unsupported Attributes response group changing  
1246 the value to the out-of-band 'unsupported' value (see section 4.1), reject the request, and return  
1247 the 'client-error-attributes-or-values-not-supported' status code. See section 3.2.1.2 for returning  
1248 unsupported attributes and values.

1249

1250 "job-k-octets" (integer(0:MAX))

1251 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports  
1252 this attribute and the "job-k-octets-supported" attribute (see section 4.4.30). The client supplied  
1253 "job-k-octets" operation attribute identifies the total size of the document(s) in K octets being  
1254 submitted (see section 4.3.17 for the complete semantics). If the client supplies the attribute and  
1255 the Printer object supports the attribute, the value of the attribute is used to populate the Job  
1256 object's "job-k-octets" Job Description attribute.

1257

1258 Note: For this attribute and the following two attributes ("job-impressions", and "job-media-  
1259 sheets"), if the client supplies the attribute, but the Printer object does not support the attribute,  
1260 the Printer object ignores the client-supplied value. If the client supplies the attribute and the  
1261 Printer supports the attribute, and the value is within the range of the corresponding Printer

1262 object's "xxx-supported" attribute, the Printer object MUST use the value to populate the Job  
1263 object's "xxx" attribute. If the client supplies the attribute and the Printer supports the attribute,  
1264 but the value is outside the range of the corresponding Printer object's "xxx-supported" attribute,  
1265 the Printer object MUST copy the attribute and its value to the Unsupported Attributes response  
1266 group, reject the request, and return the 'client-error-attributes-or-values-not-supported' status  
1267 code. If the client does not supply the attribute, the Printer object MAY choose to populate the  
1268 corresponding Job object attribute depending on whether the Printer object supports the attribute  
1269 and is able to calculate or discern the correct value.

1270

1271 "job-impressions" (integer(0:MAX))

1272 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports  
1273 this attribute and the "job-impressions-supported" attribute (see section 4.4.31). The client  
1274 supplied "job-impressions" operation attribute identifies the total size in number of impressions  
1275 of the document(s) being submitted (see section 4.3.18 for the complete semantics).

1276

1277 See note under "job-k-octets".

1278

1279 "job-media-sheets" (integer(0:MAX))

1280 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports  
1281 this attribute and the "job-media-sheets-supported" attribute (see section 4.4.32). The client  
1282 supplied "job-media-sheets" operation attribute identifies the total number of media sheets to be  
1283 produced for this job (see section 4.3.19 for the complete semantics).

1284

1285 See note under "job-k-octets".

1286

## 1287 Group 2: Job Template Attributes

1288 The client OPTIONALLY supplies a set of Job Template attributes as defined in section 4.2. If  
1289 the client is not supplying any Job Template attributes in the request, the client SHOULD omit  
1290 Group 2 rather than sending an empty group. However, a Printer object MUST be able to accept  
1291 an empty group.

1292

## 1293 Group 3: Document Content

1294 The client MUST supply the document data to be processed.

1295

1296 Note: In addition to the MANDATORY parameters required for every operation request, the simplest  
1297 Print-Job Request consists of just the "attributes-charset" and "attributes-natural-language" operation  
1298 attributes; the "printer-uri" target operation attribute; the Document Content and nothing else. In this  
1299 simple case, the Printer object:

- 1300 - creates a new Job object (the Job object contains a single document),
- 1301 - stores a generated Job name in the "job-name" attribute in the natural language and charset  
1302 requested (see Section 3.1.4.1) (if those are supported, otherwise using the Printer object's default  
1303 natural language and charset), and

1304 - at job processing time, uses its corresponding default value attributes for the supported Job  
1305 Template attributes that were not supplied by the client as IPP attribute or embedded instructions  
1306 in the document data.  
1307

### 1308 3.2.1.2 Print-Job Response

1309 The Printer object **MUST** return to the client the following sets of attributes as part of the Print-Job  
1310 Response:

#### 1311 Group 1: Operation Attributes

##### 1312 Status Message:

1313 In addition to the **REQUIRED** status code returned in every response, the response  
1314 **OPTIONALLY** includes a "status-message" (text) operation attribute as described in sections 14  
1315 and 3.1.6. If the client supplies unsupported or conflicting Job Template attributes or values, the  
1316 Printer object **MUST** reject or accept the Print-Job request depending on the whether the client  
1317 supplied a 'true' or 'false' value for the "ipp-attribute-fidelity" operation attribute. See the  
1318 Implementer's Guide [IPP-IIG] for a complete description of the suggested steps for processing a  
1319 create request.  
1320

##### 1321 Natural Language and Character Set:

1322 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1323 3.1.4.2.  
1324

#### 1325 Group 2: Unsupported Attributes

1326 This is a set of Operation and Job Template attributes supplied by the client (in the request) that  
1327 are not supported by the Printer object or that conflict with one another (see the Implementer's  
1328 Guide [IPP-IIG]). If the Printer object is not returning any Unsupported Attributes in the  
1329 response, the Printer object **SHOULD** omit Group 2 rather than sending an empty group.  
1330 However, a client **MUST** be able to accept an empty group.  
1331

1332 Unsupported attributes fall into three categories:

- 1333 1. The Printer object does not support the supplied attribute (no matter what the attribute syntax  
1334 or value).
- 1335 2. The Printer object does support the attribute, but does not support some or all of the particular  
1336 attribute syntaxes or values supplied by the client (i.e., the Printer object does not have  
1337 those attribute syntaxes or values in its corresponding "xxx-supported" attribute).
- 1338 3. The Printer object does support the attributes and values supplied, but the particular values are  
1339 in conflict with one another, because they violate a constraint, such as not being able to  
1340 staple transparencies.  
1341  
1342

1343 In the case of an unsupported attribute name, the Printer object returns the client-supplied  
1344 attribute with a substituted "out-of-band" value of 'unsupported' indicating no support for the  
1345 attribute itself (see the beginning of section 4.1).

1346  
1347 In the case of a supported attribute with one or more unsupported attribute syntaxes or values, the  
1348 Printer object simply returns the client-supplied attribute with the unsupported attribute syntaxes  
1349 or values as supplied by the client. This indicates support for the attribute, but no support for that  
1350 particular attribute syntax or value. If the client supplies a multi-valued attribute with more than  
1351 one value and the Printer object supports the attribute but only supports a subset of the client-  
1352 supplied attribute syntaxes or values, the Printer object MUST return only those attribute  
1353 syntaxes or values that are unsupported.

1354  
1355 In the case of two (or more) supported attribute values that are in conflict with one another  
1356 (although each is supported independently, the values conflict when requested together within the  
1357 same job), the Printer object MUST return all the values that it ignores or substitutes to resolve  
1358 the conflict, but not any of the values that it is still using. The choice for exactly how to resolve  
1359 the conflict is implementation dependent. See The Implementer's Guide [IPP-IIG] for an  
1360 example.

1361  
1362 In these three cases, the value of the "ipp-attribute-fidelity" supplied by the client does not affect  
1363 what the Printer object returns. The value of "ipp-attribute-fidelity" only affects whether the  
1364 Print-Job operation is accepted or rejected. If the job is accepted, the client may query the job  
1365 using the Get-Job-Attributes operation requesting the unsupported attributes that were returned in  
1366 the create response to see which attributes were ignored (not stored on the Job object) and which  
1367 attributes were stored with other (substituted) values.

1368

### 1369 Group 3: Job Object Attributes

1370 "job-uri" (uri):

1371 The Printer object MUST return the Job object's URI by returning the contents of the  
1372 REQUIRED "job-uri" Job object attribute. The client uses the Job object's URI when directing  
1373 operations at the Job object. The Printer object always uses its configured security policy when  
1374 creating the new URI. However, if the Printer object supports more than one URI, the Printer  
1375 object also uses information about which URI was used in the Print-Job Request to generate the  
1376 new URI so that the new URI references the correct access channel. In other words, if the Print-  
1377 Job Request comes in over a secure channel, the Printer object MUST generate a Job URI that  
1378 uses the secure channel as well.

1379

1380 "job-id" (integer(1:MAX)):

1381 The Printer object MUST return the Job object's Job ID by returning the REQUIRED "job-id"  
1382 Job object attribute. The client uses this "job-id" attribute in conjunction with the "printer-uri"  
1383 attribute used in the Print-Job Request when directing Job operations at the Printer object.

1384

1385 "job-state":  
1386 The Printer object **MUST** return the Job object's **REQUIRED** "job-state" attribute. The value of  
1387 this attribute (along with the value of the next attribute "job-state-reasons") is taken from a  
1388 "snapshot" of the new Job object at some meaningful point in time (implementation defined)  
1389 between when the Printer object receives the Print-Job Request and when the Printer object  
1390 returns the response.

1391  
1392 "job-state-reasons":  
1393 The Printer object **OPTIONALLY** returns the Job object's **OPTIONAL** "job-state-reasons"  
1394 attribute. If the Printer object supports this attribute then it **MUST** be returned in the response. If  
1395 this attribute is not returned in the response, the client can assume that the "job-state-reasons"  
1396 attribute is not supported and will not be returned in a subsequent Job object query.

1397  
1398 "job-state-message":  
1399 The Printer object **OPTIONALLY** returns the Job object's **OPTIONAL** "job-state-message"  
1400 attribute. If the Printer object supports this attribute then it **MUST** be returned in the response. If  
1401 this attribute is not returned in the response, the client can assume that the "job-state-message"  
1402 attribute is not supported and will not be returned in a subsequent Job object query.

1403  
1404 "number-of-intervening-jobs":  
1405 The Printer object **OPTIONALLY** returns the Job object's **OPTIONAL** "number-of-intervening-  
1406 jobs" attribute. If the Printer object supports this attribute then it **MUST** be returned in the  
1407 response. If this attribute is not returned in the response, the client can assume that the "number-  
1408 of-intervening-jobs" attribute is not supported and will not be returned in a subsequent Job object  
1409 query.

1410  
1411 Note: Since any printer state information which affects a job's state is reflected in the "job-state"  
1412 and "job-state-reasons" attributes, it is sufficient to return only these attributes and no specific  
1413 printer status attributes.

1414  
1415 Note: In addition to the **MANDATORY** parameters required for every operation response, the simplest  
1416 response consists of the just the "attributes-charset" and "attributes-natural-language" operation  
1417 attributes and the "job-uri", "job-id", and "job-state" Job Object Attributes. In this simplest case, the  
1418 status code is "successful-ok" and there is no "status-message" operation attribute.

### 1419 3.2.2 Print-URI Operation

1420 This **OPTIONAL** operation is identical to the Print-Job operation (section 3.2.1) except that a client  
1421 supplies a URI reference to the document data using the "document-uri" (uri) operation attribute (in  
1422 Group 1) rather than including the document data itself. Before returning the response, the Printer  
1423 **MUST** validate that the Printer supports the retrieval method (e.g., http, ftp, etc.) implied by the URI,  
1424 and **MUST** check for valid URI syntax. If the client-supplied URI scheme is not supported, i.e. the value  
1425 is not in the Printer object's "referenced-uri-scheme-supported" attribute, the Printer object **MUST** reject  
1426 the request and return the 'client-error-uri-scheme-not-supported' status code. See The Implementer's

1427 Guide [IPP-IIG] for suggested additional checks. The Printer NEED NOT follow the reference and  
1428 validate the contents of the reference.

1429 If the Printer object supports this operation, it MUST support the "reference-uri-schemes-supported"  
1430 Printer attribute (see section 4.4.24).

1431 It is up to the IPP object to interpret the URI and subsequently "pull" the document from the source  
1432 referenced by the URI string.

### 1433 3.2.3 Validate-Job Operation

1434 This REQUIRED operation is similar to the Print-Job operation (section 3.2.1) except that a client  
1435 supplies no document data and the Printer allocates no resources (i.e., it does not create a new Job  
1436 object). This operation is used only to verify capabilities of a printer object against whatever attributes  
1437 are supplied by the client in the Validate-Job request. By using the Validate-Job operation a client can  
1438 validate that an identical Print-Job operation (with the document data) would be accepted. The Validate-  
1439 Job operation also performs the same security negotiation as the Print-Job operation (see section 8), so  
1440 that a client can check that the client and Printer object security requirements can be met before  
1441 performing a Print-Job operation.

1442 Note: The Validate-Job operation does not accept a "document-uri" attribute in order to allow a client to  
1443 check that the same Print-URI operation will be accepted, since the client doesn't send the data with the  
1444 Print-URI operation. The client SHOULD just issue the Print-URI request.

1445 The Printer object returns the same status codes, Operation Attributes (Group 1) and Unsupported  
1446 Attributes (Group 2) as the Print-Job operation. However, no Job Object Attributes (Group 3) are  
1447 returned, since no Job object is created.

### 1448 3.2.4 Create-Job Operation

1449 This OPTIONAL operation is similar to the Print-Job operation (section 3.2.1) except that in the Create-  
1450 Job request, a client does not supply document data or any reference to document data. Also, the client  
1451 does not supply any of the "document-name", "document-format", "compression", or "document-natural-  
1452 language" operation attributes. This operation is followed by one or more Send-Document or Send-URI  
1453 operations. In each of those operation requests, the client OPTIONALLY supplies the "document-  
1454 name", "document-format", and "document-natural-language" attributes for each document in the multi-  
1455 document Job object.

1456 If a Printer object supports the Create-Job operation, it MUST also support the Send-Document  
1457 operation and also MAY support the Send-URI operation.

1458 If the Printer object supports this operation, it MUST support the "multiple-operation-time-out" Printer  
1459 attribute (see section 4.4.28).

## 1460 3.2.5 Get-Printer-Attributes Operation

1461 This REQUIRED operation allows a client to request the values of the attributes of a Printer object. In  
1462 the request, the client supplies the set of Printer attribute names and/or attribute group names in which  
1463 the requester is interested. In the response, the Printer object returns a corresponding attribute set with  
1464 the appropriate attribute values filled in.

1465 For Printer objects, the possible names of attribute groups are:

- 1466 - 'job-template': all of the Job Template attributes that apply to a Printer object (the last two columns  
1467 of the table in Section 4.2).
  - 1468 - 'printer-description': the attributes specified in Section 4.4.
  - 1469 - 'all': the special group 'all' that includes all supported attributes.
- 1470

1471 Since a client MAY request specific attributes or named groups, there is a potential that there is some  
1472 overlap. For example, if a client requests, 'printer-name' and 'all', the client is actually requesting the  
1473 "printer-name" attribute twice: once by naming it explicitly, and once by inclusion in the 'all' group. In  
1474 such cases, the Printer object NEED NOT return each attribute only once in the response even if it is  
1475 requested multiple times. The client SHOULD NOT request the same attribute in multiple ways.

1476 It is NOT REQUIRED that a Printer object support all attributes belonging to a group (since some  
1477 attributes are OPTIONAL). However, it is REQUIRED that each Printer object support all group names.

## 1478 3.2.5.1 Get-Printer-Attributes Request

1479 The following sets of attributes are part of the Get-Printer-Attributes Request:

## 1480 Group 1: Operation Attributes

1481 Natural Language and Character Set:

1482 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1483 3.1.4.1.

1484

1485 Target:

1486 The "printer-uri" (uri) operation attribute which is the target for this operation as described in  
1487 section 3.1.5.

1488

1489 Requesting User Name:

1490 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1491 described in section 8.3.

1492

1493 "requested-attributes" (1setOf keyword) :

1494 The client OPTIONALLY supplies a set of attribute names and/or attribute group names in  
1495 whose values the requester is interested. The Printer object MUST support this attribute. If the  
1496 client omits this attribute, the Printer MUST respond as if this attribute had been supplied with a  
1497 value of 'all'.

1498

1499 "document-format" (mimeMediaType) :

1500 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1501 attribute. This attribute is useful for a Printer object to determine the set of supported attribute  
1502 values that relate to the requested document format. The Printer object MUST return the  
1503 attributes and values that it uses to validate a job on a create or Validate-Job operation in which  
1504 this document format is supplied. The Printer object SHOULD return only (1) those attributes  
1505 that are supported for the specified format and (2) the attribute values that are supported for the  
1506 specified document format. By specifying the document format, the client can get the Printer  
1507 object to eliminate the attributes and values that are not supported for a specific document  
1508 format. For example, a Printer object might have multiple interpreters to support both  
1509 'application/postscript' (for PostScript) and 'text/plain' (for text) documents. However, for only  
1510 one of those interpreters might the Printer object be able to support "number-up" with values of  
1511 '1', '2', and '4'. For the other interpreter it might be able to only support "number-up" with a value  
1512 of '1'. Thus a client can use the Get-Printer-Attributes operation to obtain the attributes and  
1513 values that will be used to accept/reject a create job operation.

1514

1515 If the Printer object does not distinguish between different sets of supported values for each  
1516 different document format when validating jobs in the create and Validate-Job operations, it  
1517 MUST NOT distinguish between different document formats in the Get-Printer-Attributes  
1518 operation. If the Printer object does distinguish between different sets of supported values for  
1519 each different document format specified by the client, this specialization applies only to the  
1520 following Printer object attributes:

1521

- 1522 - Printer attributes that are Job Template attributes ("xxx-default" "xxx-supported", and "xxx-  
1523 ready" in the Table in Section 4.2),
- 1524 - "pdl-override-supported",
- 1525 - "compression-supported",
- 1526 - "job-k-octets-supported",
- 1527 - "job-impressions-supported",
- 1528 - "job-media-sheets-supported"
- 1529 - "printer-driver-installer",
- 1530 - "color-supported", and
- 1531 - "reference-uri-schemes-supported"

1532

1533 The values of all other Printer object attributes (including "document-format-supported") remain  
1534 invariant with respect to the client supplied document format (except for new Printer description  
1535 attribute as registered according to section 6.2).

1536

1537 If the client omits this "document-format" operation attribute, the Printer object MUST respond  
1538 as if the attribute had been supplied with the value of the Printer object's "document-format-  
1539 default" attribute. It is recommended that the client always supply a value for "document-  
1540 format", since the Printer object's "document-format-default" may be 'application/octet-stream',  
1541 in which case the returned attributes and values are for the union of the document formats that

1542 the Printer can automatically sense. For more details, see the description of the  
1543 'mimeMediaType' attribute syntax in section 4.1.9.

1544  
1545 If the client supplies a value for the "document-format" Operation attribute that is not supported  
1546 by the Printer, i.e., is not among the values of the Printer object's "document-format-supported"  
1547 attribute, the Printer object MUST reject the operation and return the 'client-error-document-  
1548 format-not-supported' status code.  
1549

### 1550 3.2.5.2 Get-Printer-Attributes Response

1551 The Printer object returns the following sets of attributes as part of the Get-Printer-Attributes Response:

#### 1552 Group 1: Operation Attributes

##### 1553 Status Message:

1554 In addition to the REQUIRED status code returned in every response, the response  
1555 OPTIONALLY includes a "status-message" (text) operation attribute as described in section  
1556 3.1.6.

##### 1557 1558 Natural Language and Character Set:

1559 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1560 3.1.4.2.  
1561

#### 1562 Group 2: Unsupported Attributes

1563 This is a set of Operation attributes supplied by the client (in the request) that are not supported  
1564 by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16). The response  
1565 NEED NOT contain the "requested-attributes" operation attribute with any supplied values  
1566 (attribute keywords) that were requested by the client but are not supported by the IPP object. If  
1567 the Printer object is not returning any Unsupported Attributes in the response, the Printer object  
1568 SHOULD omit Group 2 rather than sending an empty group. However, a client MUST be able  
1569 to accept an empty group.  
1570

#### 1571 Group 3: Printer Object Attributes

1572 This is the set of requested attributes and their current values. The Printer object ignores (does  
1573 not respond with) any requested attribute which is not supported. The Printer object MAY  
1574 respond with a subset of the supported attributes and values, depending on the security policy in  
1575 force. However, the Printer object MUST respond with the 'unknown' value for any supported  
1576 attribute (including all REQUIRED attributes) for which the Printer object does not know the  
1577 value. Also the Printer object MUST respond with the 'no-value' for any supported attribute  
1578 (including all REQUIRED attributes) for which the system administrator has not configured a  
1579 value. See the description of the "out-of-band" values in the beginning of Section 4.1.  
1580

## 1581 3.2.6 Get-Jobs Operation

1582 This REQUIRED operation allows a client to retrieve the list of Job objects belonging to the target  
1583 Printer object. The client may also supply a list of Job attribute names and/or attribute group names. A  
1584 group of Job object attributes will be returned for each Job object that is returned.

1585 This operation is similar to the Get-Job-Attributes operation, except that this Get-Jobs operation returns  
1586 attributes from possibly more than one object (see the description of Job attribute group names in section  
1587 3.3.4).

## 1588 3.2.6.1 Get-Jobs Request

1589 The client submits the Get-Jobs request to a Printer object.

1590 The following groups of attributes are part of the Get-Jobs Request:

## 1591 Group 1: Operation Attributes

## 1592 Natural Language and Character Set:

1593 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1594 3.1.4.1.

1595

## 1596 Target:

1597 The "printer-uri" (uri) operation attribute which is the target for this operation as described in  
1598 section 3.1.5.

1599

## 1600 Requesting User Name:

1601 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1602 described in section 8.3.

1603

## 1604 "limit" (integer(1:MAX)):

1605 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1606 attribute. It is an integer value that indicates a limit to the number of Job objects returned. The  
1607 limit is a "stateless limit" in that if the value supplied by the client is 'N', then only the first 'N'  
1608 jobs are returned in the Get-Jobs Response. There is no mechanism to allow for the next 'M' jobs  
1609 after the first 'N' jobs. If the client does not supply this attribute, the Printer object responds with  
1610 all applicable jobs.

1611

## 1612 "requested-attributes" (1setOf keyword):

1613 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1614 attribute. It is a set of Job attribute names and/or attribute groups names in whose values the  
1615 requester is interested. This set of attributes is returned for each Job object that is returned. The  
1616 allowed attribute group names are the same as those defined in the Get-Job-Attributes operation  
1617 in section 3.3.4. If the client does not supply this attribute, the Printer MUST respond as if the  
1618 client had supplied this attribute with two values: 'job-uri' and 'job-id'.

1619

1620 "which-jobs" (keyword):

1621 The client **OPTIONALLY** supplies this attribute. The Printer object **MUST** support this  
1622 attribute. It indicates which Job objects **MUST** be returned by the Printer object. The values for  
1623 this attribute are:

1624

1625 'completed': This includes any Job object whose state is 'completed', 'canceled', or 'aborted'.

1626 'not-completed': This includes any Job object whose state is 'pending', 'processing',  
1627 'processing-stopped', or 'pending-held'.

1628

1629 A Printer object **MUST** support both values. However, if the implementation does not keep jobs  
1630 in the 'completed', 'canceled', and 'aborted' states, then it returns no jobs when the 'completed'  
1631 value is supplied.

1632

1633 If a client supplies some other value, the Printer object **MUST** copy the attribute and the  
1634 unsupported value to the Unsupported Attributes response group, reject the request, and return  
1635 the 'client-error-attributes-or-values-not-supported' status code.

1636

1637 If the client does not supply this attribute, the Printer object **MUST** respond as if the client had  
1638 supplied the attribute with a value of 'not-completed'.

1639

1640 "my-jobs" (boolean):

1641 The client **OPTIONALLY** supplies this attribute. The Printer object **MUST** support this  
1642 attribute. It indicates whether all jobs or just the jobs submitted by the requesting user of this  
1643 request **MUST** be returned by the Printer object. If the client does not supply this attribute, the  
1644 Printer object **MUST** respond as if the client had supplied the attribute with a value of 'false', i.e.,  
1645 all jobs. The means for authenticating the requesting user and matching the jobs is described in  
1646 section 8.

### 1647 3.2.6.2 Get-Jobs Response

1648 The Printer object returns all of the Job objects that match the criteria as defined by the attribute values  
1649 supplied by the client in the request. It is possible that no Job objects are returned since there may  
1650 literally be no Job objects at the Printer, or there may be no Job objects that match the criteria supplied  
1651 by the client. If the client requests any Job attributes at all, there is a set of Job Object Attributes  
1652 returned for each Job object.

#### 1653 Group 1: Operation Attributes

1654 Status Message:

1655 In addition to the **REQUIRED** status code returned in every response, the response  
1656 **OPTIONALLY** includes a "status-message" (text) operation attribute as described in sections 14  
1657 and 3.1.6.

1658

1659 Natural Language and Character Set:

1660 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1661 3.1.4.2.

1662

## 1663 Group 2: Unsupported Attributes

1664 This is a set of Operation attributes supplied by the client (in the request) that are not supported  
1665 by the Printer object or that conflict with one another (see sections 3.2.1.2 and the Implementer's  
1666 Guide [IPP-IIG]). The response NEED NOT contain the "requested-attributes" operation  
1667 attribute with any supplied values (attribute keywords) that were requested by the client but are  
1668 not supported by the IPP object. If the Printer object is not returning any Unsupported Attributes  
1669 in the response, the Printer object SHOULD omit Group 2 rather than sending an empty group.  
1670 However, a client MUST be able to accept an empty group.

1671

## 1672 Groups 3 to N: Job Object Attributes

1673 The Printer object responds with one set of Job Object Attributes for each returned Job object.  
1674 The Printer object ignores (does not respond with) any requested attribute or value which is not  
1675 supported or which is restricted by the security policy in force, including whether the requesting  
1676 user is the user that submitted the job (job originating user) or not (see section 8). However, the  
1677 Printer object MUST respond with the 'unknown' value for any supported attribute (including all  
1678 REQUIRED attributes) for which the Printer object does not know the value, unless it would  
1679 violate the security policy. See the description of the "out-of-band" values in the beginning of  
1680 Section 4.1.

1681

1682 Jobs are returned in the following order:

- 1683 - If the client requests all 'completed' Jobs (Jobs in the 'completed', 'aborted', or 'canceled'  
1684 states), then the Jobs are returned newest to oldest (with respect to actual completion  
1685 time)
- 1686 - If the client requests all 'not-completed' Jobs (Jobs in the 'pending', 'processing', 'pending-  
1687 held', and 'processing-stopped' states), then Jobs are returned in relative chronological  
1688 order of expected time to complete (based on whatever scheduling algorithm is  
1689 configured for the Printer object).

1690

## 1691 3.2.7 Pause-Printer Operation

1692 This OPTIONAL operation allows a client to stop the Printer object from scheduling jobs on all its  
1693 devices. Depending on implementation, the Pause-Printer operation MAY also stop the Printer from  
1694 processing the current job or jobs. Any job that is currently being printed is either stopped as soon as the  
1695 implementation permits or is completed, depending on implementation. The Printer object MUST still  
1696 accept create operations to create new jobs, but MUST prevent any jobs from entering the 'processing'  
1697 state.

1698 If the Pause-Printer operation is supported, then the Resume-Printer operation MUST be supported, and  
1699 vice-versa.

1700 The IPP Printer stops the current job(s) on its device(s) that were in the 'processing' or 'processing-  
 1701 stopped' states as soon as the implementation permits. If the implementation supports the "printer-state-  
 1702 reasons" attribute and the devices will take appreciable time to stop, the IPP Printer adds the 'moving-to-  
 1703 paused' value to the Printer object's "printer-state-reasons" attribute (see section 4.4.11). When the  
 1704 device(s) have all stopped, the IPP Printer transitions the Printer object to the 'stopped' state, removes the  
 1705 'moving-to-paused' value, if present, and adds the 'paused' value to the Printer object's "printer-state-  
 1706 reasons" attribute.

1707 When the current job(s) complete that were in the 'processing' state, the IPP Printer transitions them to  
 1708 the 'completed' state. When the current job(s) stop in mid processing that were in the 'processing' state,  
 1709 the IPP Printer transitions them to the 'processing-stopped' state and, if the "job-state-reasons" attribute is  
 1710 supported, adds the 'printer-stopped' value to the job's "job-state-reasons" attribute.

1711 Note: for any jobs that are 'pending' or 'pending-held', the 'printer-stopped' value of the jobs' "job-state-  
 1712 reasons" attribute also applies. However, the IPP Printer NEED NOT update those jobs' "job-state-  
 1713 reasons" attributes and only need return the 'printer-stopped' value when those jobs are queried (so-called  
 1714 "lazy evaluation").

1715 Whether the Pause-Printer operation affects jobs that were submitted to the device from other sources  
 1716 than the IPP Printer object in the same way that the Pause-Printer operation affects jobs that were  
 1717 submitted to the IPP Printer object using IPP, depends on implementation, i.e., on whether the IPP  
 1718 protocol is being used as a universal management protocol or just to manage IPP jobs, respectively.

1719 The IPP Printer MUST accept the request in any state and transition the Printer to the indicated new  
 1720 "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'

1721 *Access Rights:* The requesting user must be an operator or administrator of the Printer object.  
 1722 Otherwise, the IPP Printer MUST reject the operation and return: 'client-error-forbidden', 'client-error-  
 1723 not-authenticated', or 'client-error-not-authorized' as appropriate.

### 1724 3.2.7.1 Pause-Printer Request

1725 The following groups of attributes are part of the Pause-Printer Request:

## 1726 Group 1: Operation Attributes

## 1727 Natural Language and Character Set:

1728 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1729 3.1.4.1.

1730

## 1731 Target:

1732 The "printer-uri" (uri) operation attribute which is the target for this operation as described in  
1733 section 3.1.5.

1734

## 1735 Requesting User Name:

1736 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1737 described in section 8.3.

## 1738 3.2.7.2 Pause-Printer Response

1739 The following groups of attributes are part of the Pause-Printer Response:

## 1740 Group 1: Operation Attributes

## 1741 Status Message:

1742 In addition to the REQUIRED status code returned in every response, the response  
1743 OPTIONALLY includes a "status-message" (text) operation attribute as described in section  
1744 3.1.6.

1745

## 1746 Natural Language and Character Set:

1747 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1748 3.1.4.2.

1749

## 1750 Group 2: Unsupported Attributes

1751 This is a set of Operation attributes supplied by the client (in the request) that are not supported  
1752 by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16).

## 1753 3.2.8 Resume-Printer Operation

1754 This operation allows a client to resume the Printer object scheduling jobs on all its devices. If the  
1755 Printer object supports the "printer-state-reasons" attribute, it MUST remove the 'paused' and 'moving-  
1756 to-paused' values from the Printer object's "printer-state-reasons" attribute, if present. If there are no  
1757 other reasons to keep a device paused (such as media-jam), the IPP Printer transitions itself to the  
1758 'processing' or 'idle' states, depending on whether there are jobs to be processed or not, respectively, and  
1759 the device(s) resume processing jobs.

1760 If the Pause-Printer operation is supported, then the Resume-Printer operation MUST be supported, and  
1761 vice-versa.

1762 The IPP Printer removes the 'printer-stopped' value from any job's "job-state-reasons" attributes  
1763 contained in that Printer.

1764 The IPP Printer **MUST** accept the request in any state, transition the Printer object to the indicated new  
 1765 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.

1766 *Access Rights:* The requesting user must be an operator or administrator of the Printer object.  
 1767 Otherwise, the IPP Printer **MUST** reject the operation and return: 'client-error-forbidden', 'client-error-  
 1768 not-authenticated', or 'client-error-not-authorized' as appropriate.

1769 The Resume-Printer Request and Resume-Printer Response have the same attribute groups and attributes  
 1770 as the Pause-Printer operation (see sections 3.2.7.1 and 3.2.7.2).

### 1771 3.2.9 Purge-Jobs Operation

1772 This **OPTIONAL** operation allows a client to remove all jobs from an IPP Printer object, regardless of  
 1773 their job states, including jobs in the Printer object's Job History (see Section 4.3.7.1). After a Purge-  
 1774 Jobs operation has been performed, a Printer object **MUST** return no jobs in subsequent Get-Job-  
 1775 Attributes and Get-Jobs responses (until new jobs are submitted).

1776 Whether the Purge-Jobs (and Get-Jobs) operation affects jobs that were submitted to the device from  
 1777 other sources than the IPP Printer object in the same way that the Purge-Jobs operation affects jobs that  
 1778 were submitted to the IPP Printer object using IPP, depends on implementation, i.e., on whether the IPP  
 1779 protocol is being used as a universal management protocol or just to manage IPP jobs, respectively.

1780 Note: if an operator wants to cancel all jobs without clearing out the Job History, the operator uses the  
 1781 Cancel-Job operation on each job instead of using the Purge-Job operation.

1782 The Printer object **MUST** accept this operation in any state and transition the Printer object to the 'idle'  
 1783 state.

1784 *Access Rights:* The requesting user must be an operator or administrator of the Printer object.  
 1785 Otherwise, the IPP object **MUST** reject the operation and return: client-error-forbidden, client-error-not-  
 1786 authenticated, and client-error-not-authorized as appropriate.

1787 The Purge-Jobs Request and Purge-Jobs Response have the same attribute groups and attributes as the  
 1788 Pause-Printer operation (see sections 3.2.7.1 and 3.2.7.2).

## 1789 3.3 Job Operations

1790 All Job operations are directed at Job objects. A client **MUST** always supply some means of identifying  
1791 the Job object in order to identify the correct target of the operation. That job identification **MAY** either  
1792 be a single Job URI or a combination of a Printer URI with a Job ID. The IPP object implementation  
1793 **MUST** support both forms of identification for every job.

## 1794 3.3.1 Send-Document Operation

1795 This **OPTIONAL** operation allows a client to create a multi-document Job object that is initially "empty"  
1796 (contains no documents). In the Create-Job response, the Printer object returns the Job object's URI (the  
1797 "job-uri" attribute) and the Job object's 32-bit identifier (the "job-id" attribute). For each new document  
1798 that the client desires to add, the client uses a Send-Document operation. Each Send-Document Request  
1799 contains the entire stream of document data for one document.

1800 Since the Create-Job and the send operations (Send-Document or Send-URI operations) that follow  
1801 could occur over an arbitrarily long period of time for a particular job, a client **MUST** send another send  
1802 operation within an IPP Printer defined minimum time interval after the receipt of the previous request  
1803 for the job. If a Printer object supports multiple document jobs, the Printer object **MUST** support the  
1804 "multiple-operation-time-out" attribute (see section 4.4.28). This attribute indicates the minimum  
1805 number of seconds the Printer object will wait for the next send operation before taking some recovery  
1806 action.

1807 An IPP object **MUST** recover from an errant client that does not supply a send operation, sometime after  
1808 the minimum time interval specified by the Printer object's "multiple-operation-time-out" attribute. Such  
1809 recovery **MAY** include any of the following or other recovery actions:

- 1810 1. Assume that the Job is an invalid job, start the process of changing the job state to 'aborted', add  
1811 the 'aborted-by-system' value to the job's "job-state-reasons" attribute (see section 4.3.8), if  
1812 supported, and clean up all resources associated with the Job. In this case, if another send  
1813 operation is finally received, the Printer responds with an "client-error-not-possible" or "client-  
1814 error-not-found" depending on whether or not the Job object is still around when the send  
1815 operation finally arrives.
- 1816 2. Assume that the last send operation received was in fact the last document (as if the "last-  
1817 document" flag had been set to 'true'), close the Job object, and proceed to process it (i.e., move  
1818 the Job's state to 'pending').
- 1819 3. Assume that the last send operation received was in fact the last document, close the Job, but  
1820 move it to the 'pending-held' and add the 'submission-interrupted' value to the job's "job-state-  
1821 reasons" attribute (see section 4.3.8), if supported. This action allows the user or an operator to  
1822 determine whether to continue processing the Job by moving it back to the 'pending' state using  
1823 the Release-Job operation (see section 3.3.6) or to cancel the job using the Cancel-Job operation  
1824 (see section 3.3.3).

1825

1826 Each implementation is free to decide the "best" action to take depending on local policy, whether any  
1827 documents have been added, whether the implementation spools jobs or not, and/or any other piece of

1828 information available to it. If the choice is to abort the Job object, it is possible that the Job object may  
1829 already have been processed to the point that some media sheet pages have been printed.

### 1830 3.3.1.1 Send-Document Request

1831 The following attribute sets are part of the Send-Document Request:

#### 1832 Group 1: Operation Attributes

##### 1833 Natural Language and Character Set:

1834 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1835 3.1.4.1.

1836

##### 1837 Target:

1838 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation  
1839 attribute(s) which define the target for this operation as described in section 3.1.5.

1840

##### 1841 Requesting User Name:

1842 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1843 described in section 8.3.

1844

##### 1845 "document-name" (name(MAX)):

1846 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1847 attribute. It contains the client supplied document name. The document name MAY be different  
1848 than the Job name. It might be helpful, but NEED NOT be unique across multiple documents in  
1849 the same Job. Typically, the client software automatically supplies the document name on behalf  
1850 of the end user by using a file name or an application generated name. See the description of the  
1851 "document-name" operation attribute in the Print-Job Request (section 3.2.1.1) for more  
1852 information about this attribute.

1853

##### 1854 "document-format" (mimeMediaType) :

1855 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1856 attribute. The value of this attribute identifies the format of the supplied document data. If the  
1857 client does not supply this attribute, the Printer object assumes that the document data is in the  
1858 format defined by the Printer object's "document-format-default" attribute. If the client supplies  
1859 this attribute, but the value is not supported by the Printer object, i.e., the value is not one of the  
1860 values of the Printer object's "document-format-supported" attribute, the Printer object MUST  
1861 reject the request and return the 'client-error-document-format-not-supported' status code.

1862

##### 1863 "document-natural-language" (naturalLanguage):

1864 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports  
1865 this attribute. This attribute specifies the natural language of the document for those document-  
1866 formats that require a specification of the natural language in order to image the document  
1867 unambiguously. There are no particular values required for the Printer object to support.

1868

1869 "compression" (type3 keyword)

1870 The client **OPTIONALLY** supplies this attribute. The Printer object **OPTIONALLY** supports  
1871 this attribute and the "compression-supported" attribute (see section 4.4.29). The client supplied  
1872 "compression" operation attribute identifies the compression algorithm used on the document  
1873 data. If the client omits this attribute, the Printer object **MUST** assume that the data is not  
1874 compressed. If the client supplies the attribute and the Printer object supports the attribute, the  
1875 Printer object **MUST** use the corresponding decompression algorithm on the document data. If  
1876 the client supplies this attribute, but the value is not supported by the Printer object, i.e., the value  
1877 is not one of the values of the Printer object's "compression-supported" attribute, the Printer  
1878 object **MUST** copy the attribute and its value to the Unsupported Attributes response group,  
1879 reject the request, and return the 'client-error-attributes-or-values-not-supported' status code.

1880

1881 "last-document" (boolean):

1882 The client **MUST** supply this attribute. The Printer object **MUST** support this attribute. It is a  
1883 boolean flag that is set to 'true' if this is the last document for the Job, 'false' otherwise.

1884

1885 Group 2: Document Content

1886 The client **MUST** supply the document data if the "last-document" flag is set to 'false'. However,  
1887 since a client might not know that the previous document sent with a Send-Document (or Send-  
1888 URI) operation was the last document (i.e., the "last-document" attribute was set to 'false'), it is  
1889 legal to send a Send-Document request with no document data where the "last-document" flag is  
1890 set to 'true'. Such a request **MUST NOT** increment the value of the Job object's "number-of-  
1891 documents" attribute, since no real document was added to the job.

1892 3.3.1.2 Send-Document Response

1893 The following sets of attributes are part of the Send-Document Response:

1894 Group 1: Operation Attributes

1895 Status Message:

1896 In addition to the **REQUIRED** status code returned in every response, the response  
1897 **OPTIONALLY** includes a "status-message" (text) operation attribute as described in sections 14  
1898 and 3.1.6.

1899

1900 Natural Language and Character Set:

1901 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1902 3.1.4.2.

1903

1904 Group 2: Unsupported Attributes

1905 This is a set of Operation attributes supplied by the client (in the request) that are not supported  
1906 by the Printer object or that conflict with one another (see sections 3.2.1.2 and the Implementer's  
1907 Guide [IPP-IIG]). If the Printer object is not returning any Unsupported Attributes in the  
1908 response, the Printer object **SHOULD** omit Group 2 rather than sending an empty group.  
1909 However, a client **MUST** be able to accept an empty group.

1910

## 1911 Group 3: Job Object Attributes

1912 This is the same set of attributes as described in the Print-Job response (see section 3.2.1.2).

1913

## 1914 3.3.2 Send-URI Operation

1915 This OPTIONAL operation is identical to the Send-Document operation (see section 3.3.1) except that a  
1916 client MUST supply a URI reference ("document-uri" operation attribute) rather than the document data  
1917 itself. If a Printer object supports this operation, clients can use both Send-URI or Send-Document  
1918 operations to add new documents to an existing multi-document Job object. However, if a client needs  
1919 to indicate that the previous Send-URI or Send-Document was the last document, the client MUST use  
1920 the Send-Document operation with no document data and the "last-document" flag set to 'true' (rather  
1921 than using a Send-URI operation with no "document-uri" operation attribute).

1922 If a Printer object supports this operation, it MUST also support the Print-URI operation (see section  
1923 3.2.2).

1924 The Printer object MUST validate the syntax and URI scheme of the supplied URI before returning a  
1925 response, just as in the Print-URI operation.

## 1926 3.3.3 Cancel-Job Operation

1927 This REQUIRED operation allows a client to cancel a Print Job from the time the job is created up to the  
1928 time it is completed, canceled, or aborted. Since a Job might already be printing by the time a Cancel-  
1929 Job is received, some media sheet pages might be printed before the job is actually terminated.

## 1930 3.3.3.1 Cancel-Job Request

1931 The following groups of attributes are part of the Cancel-Job Request:

## 1932 Group 1: Operation Attributes

1933 Natural Language and Character Set:

1934 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1935 3.1.4.1.

1936

1937 Target:

1938 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri)  
1939 operation attribute(s) which define the target for this operation as described in section 3.1.5.

1940

1941 Requesting User Name:

1942 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1943 described in section 8.3.

1944

1945 "message" (text(127)):

1946 The client **OPTIONALLY** supplies this attribute. The Printer object **OPTIONALLY** supports  
1947 this attribute. It is a message to the operator. This "message" attribute is not the same as the "job-  
1948 message-from-operator" attribute. That attribute is used to report a message from the operator to  
1949 the end user that queries that attribute. This "message" operation attribute is used to send a  
1950 message from the client to the operator along with the operation request. It is an implementation  
1951 decision of how or where to display this message to the operator (if at all).  
1952

### 1953 3.3.3.2 Cancel-Job Response

1954 The following sets of attributes are part of the Cancel-Job Response:

#### 1955 Group 1: Operation Attributes

##### 1956 Status Message:

1957 In addition to the **REQUIRED** status code returned in every response, the response  
1958 **OPTIONALLY** includes a "status-message" (text) operation attribute as described in sections 14  
1959 and 3.1.6.

1960  
1961 If the job is already in the 'completed', 'aborted', or 'canceled' state, or the 'process-to-stop-point'  
1962 value is set in the Job's "job-state-reasons" attribute, the Printer object **MUST** reject the request  
1963 and return the 'client-error-not-possible' error status code.  
1964

##### 1965 Natural Language and Character Set:

1966 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1967 3.1.4.2.  
1968

#### 1969 Group 2: Unsupported Attributes

1970 This is a set of Operation attributes supplied by the client (in the request) that are not supported  
1971 by the Printer object or that conflict with one another (see section 3.2.1.2 and the Implementer's  
1972 Guide [IPP-IIG]). If the Printer object is not returning any Unsupported Attributes in the  
1973 response, the Printer object **SHOULD** omit Group 2 rather than sending an empty group.  
1974 However, a client **MUST** be able to accept an empty group.  
1975

1976 Once a successful response has been sent, the implementation guarantees that the Job will eventually end  
1977 up in the 'canceled' state. Between the time of the Cancel-Job operation is accepted and when the job  
1978 enters the 'canceled' job-state (see section 4.3.7), the "job-state-reasons" attribute **SHOULD** contain the '  
1979 processing-to-stop-point' value which indicates to later queries that although the Job might still be  
1980 'processing', it will eventually end up in the 'canceled' state, not the 'completed' state.

## 1981 3.3.4 Get-Job-Attributes Operation

1982 This REQUIRED operation allows a client to request the values of attributes of a Job object and it is  
1983 almost identical to the Get-Printer-Attributes operation (see section 3.2.5). The only differences are that  
1984 the operation is directed at a Job object rather than a Printer object, there is no "document-format"  
1985 operation attribute used when querying a Job object, and the returned attribute group is a set of Job  
1986 object attributes rather than a set of Printer object attributes.

1987 For Jobs, the possible names of attribute groups are:

- 1988 - 'job-template': all of the Job Template attributes that apply to a Job object (the first column of the  
1989 table in Section 4.2).
  - 1990 - 'job-description': all of the Job Description attributes specified in Section 4.3.
  - 1991 - 'all': the special group 'all' that includes all supported attributes.
- 1992

1993 Since a client MAY request specific attributes or named groups, there is a potential that there is some  
1994 overlap. For example, if a client requests, 'job-name' and 'job-description', the client is actually  
1995 requesting the "job-name" attribute once by naming it explicitly, and once by inclusion in the 'job-  
1996 description' group. In such cases, the Printer object NEED NOT return the attribute only once in the  
1997 response even if it is requested multiple times. The client SHOULD NOT request the same attribute in  
1998 multiple ways.

1999 It is NOT REQUIRED that a Job object support all attributes belonging to a group (since some attributes  
2000 are OPTIONAL). However it is REQUIRED that each Job object support all group names.

## 2001 3.3.4.1 Get-Job-Attributes Request

2002 The following groups of attributes are part of the Get-Job-Attributes Request when the request is  
2003 directed at a Job object:

## 2004 Group 1: Operation Attributes

2005 Natural Language and Character Set:

2006 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
2007 3.1.4.1.

2008

2009 Target:

2010 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri)  
2011 operation attribute(s) which define the target for this operation as described in section 3.1.5.

2012

2013 Requesting User Name:

2014 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
2015 described in section 8.3.

2016

2017 "requested-attributes" (1setOf keyword) :  
2018 The client **OPTIONALLY** supplies this attribute. The IPP object **MUST** support this attribute.  
2019 It is a set of attribute names and/or attribute group names in whose values the requester is  
2020 interested. If the client omits this attribute, the IPP object **MUST** respond as if this attribute had  
2021 been supplied with a value of 'all'.  
2022

#### 2023 3.3.4.2 Get-Job-Attributes Response

2024 The Printer object returns the following sets of attributes as part of the Get-Job-Attributes Response:

##### 2025 Group 1: Operation Attributes

###### 2026 Status Message:

2027 In addition to the **REQUIRED** status code returned in every response, the response  
2028 **OPTIONALLY** includes a "status-message" (text) operation attribute as described in sections 14  
2029 and 3.1.6.  
2030

###### 2031 Natural Language and Character Set:

2032 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
2033 3.1.4.2. The "attributes-natural-language" **MAY** be the natural language of the Job object, rather  
2034 than the one requested.  
2035

##### 2036 Group 2: Unsupported Attributes

2037 This is a set of Operation attributes supplied by the client (in the request) that are not supported  
2038 by the Printer object or that conflict with one another (see sections 3.2.1.2 and the Implementer's  
2039 Guide [IPP-IIG]). The response **NEED NOT** contain the "requested-attributes" operation  
2040 attribute with any supplied values (attribute keywords) that were requested by the client but are  
2041 not supported by the IPP object. If the Printer object is not returning any Unsupported Attributes  
2042 in the response, the Printer object **SHOULD** omit Group 2 rather than sending an empty group.  
2043 However, a client **MUST** be able to accept an empty group.  
2044

##### 2045 Group 3: Job Object Attributes

2046 This is the set of requested attributes and their current values. The IPP object ignores (does not  
2047 respond with) any requested attribute or value which is not supported or which is restricted by the  
2048 security policy in force, including whether the requesting user is the user that submitted the job  
2049 (job originating user) or not (see section 8). However, the IPP object **MUST** respond with the  
2050 'unknown' value for any supported attribute (including all **REQUIRED** attributes) for which the  
2051 IPP object does not know the value, unless it would violate the security policy. See the  
2052 description of the "out-of-band" values in the beginning of Section 4.1.

## 2053 3.3.5 Hold-Job Operation

2054 This OPTIONAL operation allows a client to hold a pending job in the queue so that it is not eligible for  
 2055 scheduling. If the Hold-Job operation is supported, then the Release-Job operation MUST be supported,  
 2056 and vice-versa. The OPTIONAL "job-hold-until" operation attribute allows a client to specify whether  
 2057 to hold the job indefinitely or until a specified time period, if supported.

2058 The IPP object MUST accept or reject the request based on the job's current state and transition the job  
 2059 to the indicated new state as follows:

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

2060 Note 1: If the OPTIONAL "job-state-reasons" attribute is supported and if the implementation supports  
 2061 multiple reasons for a job to be in the 'pending-held' state, the IPP object MUST add the 'job-hold-until-  
 2062 specified' value to the job's "job-state-reasons" attribute.

2063 Note 2: If the IPP object supports the "job-hold-until" operation attribute, but the specified time period  
 2064 has already started (or is the 'no-hold' value) and there are no other reasons to hold the job, the IPP object  
 2065 MUST make the job be a candidate for processing immediately (see Section 4.2.2) by putting the job in  
 2066 the 'pending' state.

2067 Note: In order to keep the Hold-Job operation simple, such a request is rejected when the job is in the  
 2068 'processing' or 'processing-stopped' states. If an operation is needed to hold jobs while in these states, it  
 2069 will be added as an additional operation, rather than overloading the Hold-Job operation. Then it is clear  
 2070 to clients by querying the Printer object's "operations-supported" (see Section 4.4.13) and the Job  
 2071 object's "job-state" (see Section 4.3.7) attributes which operations are possible.

2072 *Access Rights:* The requesting user must either be the submitter of the job or an operator or administrator  
 2073 of the Printer object (see Section 1). Otherwise, the IPP object MUST reject the operation and return:  
 2074 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' as appropriate.

## 2075 3.3.5.1 Hold-Job Request

2076 The groups and operation attributes are the same as for a Cancel-Job request (see section 3.3.3.1), with  
 2077 the addition of the following Group 1 Operation attribute:

2078 "job-hold-until" (type3 keyword | name(MAX)):  
2079 The client **OPTIONALLY** supplies this Operation attribute. The IPP object **MUST** support this  
2080 operation attribute in a Hold-Job request, if it supports the "job-hold-until" Job template attribute  
2081 in create operations. See section 4.2.2. The IPP object **SHOULD** support the "job-hold-until"  
2082 Job Template attribute for use in job create operations with at least the 'indefinite' value, if it  
2083 supports the Hold-Job operation. Otherwise, a client cannot create a job and hold it immediately  
2084 (without picking some supported time period in the future).

2085 If supplied and supported as specified in the Printer's "job-hold-until-supported" attribute, the IPP  
2086 object copies the supplied operation attribute to the Job object, replacing the job's previous "job-  
2087 hold-until" attribute, if present, and makes the job a candidate for scheduling during the supplied  
2088 named time period.

2089 If supplied, but either the "job-hold-until" Operation attribute itself or the value supplied is not  
2090 supported, the IPP object accepts the request, returns the unsupported attribute or value in the  
2091 Unsupported Attributes Group according to section 3.2.1.2, returns the 'successful-ok-ignored-or-  
2092 substituted-attributes, and holds the job indefinitely until a client performs a subsequent Release-  
2093 Job operation.

2094 If the client (1) supplies a value that specifies a time period that has already started or the 'no-  
2095 hold' value (meaning don't hold the job) and (2) the IPP object supports the "job-hold-until"  
2096 operation attribute and there are no other reasons to hold the job, the IPP object **MUST** accept the  
2097 operation and make the job be a candidate for processing immediately (see Section 4.2.2).

2098 If the client does not supply a "job-hold-until" Operation attribute in the request, the IPP object  
2099 **MUST** populate the job object with a "job-hold-until" attribute with the 'indefinite' value (if IPP  
2100 object supports the "job-hold-until" attribute) and hold the job indefinitely, until a client performs  
2101 a Release-Job operation.

### 2102 3.3.5.2 Hold-Job Response

2103 The groups and attributes are the same as for a Cancel-Job response (see section 3.3.3.2).

### 2104 3.3.6 Release-Job Operation

2105 This **OPTIONAL** operation allows a client to release a previously held job so that it is again eligible for  
2106 scheduling. If the Hold-Job operation is supported, then the Release-Job operation **MUST** be supported,  
2107 and vice-versa.

2108 This operation removes the "job-hold-until" job attribute, if present, from the job object that had been  
2109 supplied in the create or most recent Hold-Job or Restart-Job operation and remove its effect on the job.  
2110 If the **OPTIONAL** "job-state-reasons" attribute is supported, the IPP object **MUST** remove the 'job-hold-  
2111 until-specified' value from the job's "job-state-reasons" attribute, if present. See section 4.3.8.

2112 The IPP object **MUST** accept or reject the request based on the job's current state and transition the job  
2113 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'

2114 Note 1: If there are other reasons to keep the job in the 'pending-held' state, such as 'resources-are-not-  
 2115 ready', the job remains in the 'pending-held' state. Thus the 'pending-held' state is not just for jobs that  
 2116 have the 'job-hold-until' applied to them, but are for any reason to keep the job from being a candidate  
 2117 for scheduling and processing, such as 'resources-are-not-ready'. See the "job-hold-until" attribute  
 2118 (section 4.2.2).

2119 *Access Rights:* The requesting user must either be the submitter of the job or an operator or administrator  
 2120 of the Printer object. Otherwise, the IPP object MUST reject the operation and return: 'client-error-  
 2121 forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' as appropriate.

2122 The Release-Job Request and Release-Job Response have the same attribute groups and attributes as the  
 2123 Cancel-Job operation (see section 3.3.3.1 and 3.3.3.2).

### 2124 3.3.7 Restart-Job Operation

2125 This OPTIONAL operation allows a client to restart a job that is retained in the queue after processing  
 2126 has completed (see section 4.3.7.1).

2127 The job is moved to the 'pending' job state and restarts at the beginning on the same IPP Printer object  
 2128 with the same attribute values. The Job Description attributes that accumulate job progress, such as  
 2129 "job-impressions-completed", "job-media-sheets-completed", and "job-k-octets-processed", MUST be  
 2130 reset to 0 so that they give an accurate record of the job from its restart point. The job object MUST  
 2131 continue to use the same "job-uri" and "job-id" attribute values.

2132 Note: If in the future an operation is needed that does not reset the job progress attributes, then a new  
 2133 operation will be defined which makes a copy of the job, assigns a new "job-uri" and "job-id" to the copy  
 2134 and resets the job progress attributes in the new copy only.

2135 The IPP object MUST accept or reject the request based on the job's current state, transition the job to  
 2136 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'.

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

2137

2138 Note 1: If the Job Retention Period has expired for the job in this state, then the IPP object rejects the  
2139 operation. See section 4.3.7.1.

2140 Note: In order to prevent a user from inadvertently restarting a job in the middle, the Restart-Job request  
2141 is rejected when the job is in the 'processing' or 'processing-stopped' states. If in the future an operation  
2142 is needed to hold or restart jobs while in these states, it will be added as an additional operation, rather  
2143 than overloading the Restart-Job operation, so that it is clear that the user intended that the current job  
2144 not be completed.

2145 *Access Rights:* The requesting user must either be the submitter of the job or an operator or administrator  
2146 of the Printer object. Otherwise, the IPP object MUST reject the operation and return: 'client-error-  
2147 forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' as appropriate.

### 2148 3.3.7.1 Restart-Job Request

2149 The groups and attributes are the same as for a Cancel-Job request (see section 3.3.3.1), with the addition  
2150 of the following Group 1 Operation attribute:

2151 "job-hold-until" (type3 keyword | name(MAX)):

2152 The client OPTIONALLY supplies this attribute. The IPP object MUST support this Operation  
2153 attribute in a Restart-Job request, if it supports the "job-hold-until" Job Template attribute in  
2154 create operations. See section 4.2.2. Otherwise, the IPP object NEED NOT support the "job-  
2155 hold-until" Operation attribute in a Restart-Job request.

2156 If supplied and supported as specified in the Printer's "job-hold-until-supported" attribute, the IPP  
2157 object copies the supplied Operation attribute to the Job object, replacing the job's previous "job-  
2158 hold-until" attribute, if present, and makes the job a candidate for scheduling during the supplied  
2159 named time period. See section 4.2.2.

2160 If supplied, but the value is not supported, the IPP object accepts the request, returns the  
2161 unsupported attribute or value in the Unsupported Attributes Group according to section 3.2.1.2,  
2162 returns the 'successful-ok-ignored-or-substituted-attributes' status code, and holds the job  
2163 indefinitely until a client performs a subsequent Release-Job operation.

2164 If supplied, but the "job-hold-until" Operation attribute itself is not supported, the IPP object  
2165 accepts the request, returns the unsupported attribute with the out-of-band 'unsupported' value in  
2166 the Unsupported Attributes Group according to section 3.2.1.2, returns the 'successful-ok-  
2167 ignored-or-substituted-attributes' status code, and restarts the job, i.e., ignores the "job-hold-  
2168 until" attribute.

2169 If the client (1) supplies a value that specifies a time period that has already started or the 'no-  
2170 hold' value (meaning don't hold the job) and (2) the IPP object supports the "job-hold-until"  
2171 operation attribute and there are no other reasons to hold the job, the IPP object makes the job a  
2172 candidate for processing immediately (see Section 4.2.2).

2173 If the client does not supply a "job-hold-until" operation attribute in the request, the IPP object  
2174 removes the "job-hold-until" attribute, if present, from the job. If there are no other reasons to  
2175 hold the job, the Restart-Job operation makes the job a candidate for processing immediately (see  
2176 Section 4.2.2).

### 2177 3.3.7.2 Restart-Job Response

2178 The groups and attributes are the same as for a Cancel-Job response (see section 3.3.3.2).

2179 Note: In the future an OPTIONAL Modify-Job operation may be specified that allows the client to  
2180 modify other attributes before releasing the restarted job.

## 2181 4. Object Attributes

2182 This section describes the attributes with their corresponding attribute syntaxes and values that are part  
2183 of the IPP model. The sections below show the objects and their associated attributes which are  
2184 included within the scope of this protocol. Many of these attributes are derived from other relevant  
2185 specifications:

2186 - Document Printing Application (DPA) [ISO10175]

2187 - RFC 1759 Printer MIB [RFC1759]

2188

2189 Each attribute is uniquely identified in this document using a "keyword" (see section 13.2.1) which is the  
2190 name of the attribute. The keyword is included in the section header describing that attribute.

2191 Note: Not only are keywords used to identify attributes, but one of the attribute syntaxes described  
2192 below is "keyword" so that some attributes have keyword values. Therefore, these attributes are defined  
2193 as having an attribute syntax that is a set of keywords.

### 2194 4.1 Attribute Syntaxes

2195 This section defines the basic attribute syntax types that all clients and IPP objects MUST be able to  
2196 accept in responses and accept in requests, respectively. Each attribute description in sections 3 and

2197 3.3.5 includes the name of attribute syntax(es) in the heading (in parentheses). A conforming  
2198 implementation of an attribute MUST include the semantics of the attribute syntax(es) so identified.  
2199 Section 6.3 describes how the protocol can be extended with new attribute syntaxes.

2200 The attribute syntaxes are specified in the following sub-sections, where the sub-section heading is the  
2201 keyword name of the attribute syntax inside the single quotes. In operation requests and responses each  
2202 attribute value MUST be represented as one of the attribute syntaxes specified in the sub-section heading  
2203 for the attribute. In addition, the value of an attribute in a response (but not in a request) MAY be one of  
2204 the "out-of-band" values. Standard "out-of-band" values are:

2205     `unknown`: The attribute is supported by the IPP object, but the value is unknown to the IPP object  
2206             for some reason.

2207     `unsupported`: The attribute is unsupported by the IPP object. This value MUST be returned only as  
2208             the value of an attribute in the Unsupported Attributes Group.

2209     `no-value`: The attribute is supported by the Printer object, but the administrator has not yet  
2210             configured a value.

2211

2212 The "Encoding and Transport" specification [IPP-PRO] defines mechanisms for passing "out-of-band"  
2213 values. All attributes in a request MUST have one or more values as defined in Sections 4.2 to 4.4.  
2214 Thus clients MUST NOT supply attributes with "out-of-band" values. All attributes in a response  
2215 MUST have one or more values as defined in Sections 4.2 to 4.4 or a single "out-of-band" value.

2216 Most attributes are defined to have a single attribute syntax. However, a few attributes (e.g., "job-sheet",  
2217 "media", "job-hold-until") are defined to have several attribute syntaxes, depending on the value. These  
2218 multiple attribute syntaxes are separated by the "|" character in the sub-section heading to indicate the  
2219 choice. Since each value MUST be tagged as to its attribute syntax in the protocol, a single-valued  
2220 attribute instance may have any one of its attribute syntaxes and a multi-valued attribute instance may  
2221 have a mixture of its defined attribute syntaxes.

#### 2222 4.1.1 `text`

2223 A text attribute is an attribute whose value is a sequence of zero or more characters encoded in a  
2224 maximum of 1023 (MAX) octets. MAX is the maximum length for each value of any text attribute.  
2225 However, if an attribute will always contain values whose maximum length is much less than MAX, the  
2226 definition of that attribute will include a qualifier that defines the maximum length for values of that  
2227 attribute. For example: the "printer-location" attribute is specified as "printer-location (text(127))". In  
2228 this case, text values for "printer-location" MUST NOT exceed 127 octets; if supplied with a longer text  
2229 string via some external interface (other than the protocol), implementations are free to truncate to this  
2230 shorter length limitation.

2231 In this specification, all text attributes are defined using the `text` syntax. However, `text` is used only for  
2232 brevity; the formal interpretation of `text` is: `textWithoutLanguage | textWithLanguage`. That is, for any  
2233 attribute defined in this specification using the `text` attribute syntax, all IPP objects and clients MUST  
2234 support both the `textWithoutLanguage` and `textWithLanguage` attribute syntaxes. However, in actual

2235 usage and protocol execution, objects and clients accept and return only one of the two syntax per  
2236 attribute. The syntax 'text' never appears "on-the-wire".

2237 Both 'textWithoutLanguage' and 'textWithLanguage' are needed to support the real world needs of  
2238 interoperability between sites and systems that use different natural languages as the basis for human  
2239 communication. Generally, one natural language applies to all text attributes in a given request or  
2240 response. The language is indicated by the "attributes-natural-language" operation attribute defined in  
2241 section 3.1.4 or "attributes-natural-language" job attribute defined in section 4.3.24, and there is no need  
2242 to identify the natural language for each text string on a value-by-value basis. In these cases, the  
2243 attribute syntax 'textWithoutLanguage' is used for text attributes. In other cases, the client needs to  
2244 supply or the Printer object needs to return a text value in a natural language that is different from the  
2245 rest of the text values in the request or response. In these cases, the client or Printer object uses the  
2246 attribute syntax 'textWithLanguage' for text attributes (this is the Natural Language Override mechanism  
2247 described in section 3.1.4).

2248 The 'textWithoutLanguage' and 'textWithLanguage' attribute syntaxes are described in more detail in the  
2249 following sections.

#### 2250 4.1.1.1 'textWithoutLanguage'

2251 The 'textWithoutLanguage' syntax indicates a value that is sequence of zero or more characters. Text  
2252 strings are encoded using the rules of some charset. The Printer object MUST support the UTF-8  
2253 charset [RFC2279] and MAY support additional charsets to represent 'text' values, provided that the  
2254 charsets are registered with IANA [IANA-CS]. See Section 4.1.7 for the specification of the 'charset'  
2255 attribute syntax, including restricted semantics and examples of charsets.

#### 2256 4.1.1.2 'textWithLanguage'

2257 The 'textWithLanguage' attribute syntax is a compound attribute syntax consisting of two parts: a  
2258 'textWithoutLanguage' part plus an additional 'naturalLanguage' (see section 4.1.8) part that overrides the  
2259 natural language in force. The 'naturalLanguage' part explicitly identifies the natural language that  
2260 applies to the text part of that value and that value alone. For any give text attribute, the  
2261 'textWithoutLanguage' part is limited to the maximum length defined for that attribute, but the  
2262 'naturalLanguage' part is always limited to 63 octets. Using the 'textWithLanguage' attribute syntax rather  
2263 than the normal 'textWithoutLanguage' syntax is the so-called Natural Language Override mechanism  
2264 and MUST be supported by all IPP objects and clients.

2265 If the attribute is multi-valued (1setOf text), then the 'textWithLanguage' attribute syntax MUST be used  
2266 to explicitly specify each attribute value whose natural language needs to be overridden. Other values in  
2267 a multi-valued 'text' attribute in a request or a response revert to the natural language of the operation  
2268 attribute.

2269 In a create request, the Printer object MUST accept and store with the Job object any natural language in  
2270 the "attributes-natural-language" operation attribute, whether the Printer object supports that natural  
2271 language or not. Furthermore, the Printer object MUST accept and store any 'textWithLanguage'

2272 attribute value, whether the Printer object supports that natural language or not. These requirements are  
2273 independent of the value of the "ipp-attribute-fidelity" operation attribute that the client MAY supply.

2274 Example: If the client supplies the "attributes-natural-language" operation attribute with the value: 'en'  
2275 indicating English, but the value of the "job-name" attribute is in French, the client MUST use the  
2276 'textWithLanguage' attribute syntax with the following two values:

2277 'fr': Natural Language Override indicating French

2278 'Rapport Mensuel': the job name in French

2279

2280 See the "Encoding and Transport" document [IPP-PRO] for a detailed example of the  
2281 'textWithLanguage' attribute syntax.

#### 2282 4.1.2 'name'

2283 This syntax type is used for user-friendly strings, such as a Printer name, that, for humans, are more  
2284 meaningful than identifiers. Names are never translated from one natural language to another. The  
2285 'name' attribute syntax is essentially the same as 'text', including the REQUIRED support of UTF-8  
2286 except that the sequence of characters is limited so that its encoded form MUST NOT exceed 255  
2287 (MAX) octets.

2288 Also like 'text', 'name' is really an abbreviated notation for either 'nameWithoutLanguage' or  
2289 'nameWithLanguage'. That is, all IPP objects and clients MUST support both the  
2290 'nameWithoutLanguage' and 'nameWithLanguage' attribute syntaxes. However, in actual usage and  
2291 protocol execution, objects and clients accept and return only one of the two syntax per attribute. The  
2292 syntax 'name' never appears "on-the-wire".

2293 Note: Only the 'text' and 'name' attribute syntaxes permit the Natural Language Override mechanism.

2294 Some attributes are defined as 'type3 keyword | name'. These attributes support values that are either  
2295 type3 keywords or names. This dual-syntax mechanism enables a site administrator to extend these  
2296 attributes to legally include values that are locally defined by the site administrator. Such names are not  
2297 registered with IANA.

##### 2298 4.1.2.1 'nameWithoutLanguage'

2299 The 'nameWithoutLanguage' syntax indicates a value that is sequence of zero or more characters so that  
2300 its encoded form does not exceed MAX octets.

##### 2301 4.1.2.2 'nameWithLanguage'

2302 The 'nameWithLanguage' attribute syntax is a compound attribute syntax consisting of two parts: a  
2303 'nameWithoutLanguage' part plus an additional 'naturalLanguage' (see section 4.1.8) part that overrides  
2304 the natural language in force. The 'naturalLanguage' part explicitly identifies the natural language that  
2305 applies to that name value and that name value alone.

2306 The 'nameWithLanguage' attribute syntax behaves the same as the 'textWithLanguage' syntax. If a name  
2307 is in a language that is different than the rest of the object or operation, then this 'nameWithLanguage'  
2308 syntax is used rather than the generic 'nameWithoutLanguage' syntax.

2309 Example: If the client supplies the "attributes-natural-language" operation attribute with the value: 'en'  
2310 indicating English, but the "printer-name" attribute is in German, the client MUST use the  
2311 'nameWithLanguage' attribute syntax as follows:

2312       'de': Natural Language Override indicating German  
2313       'Farbdrucker': the Printer name in German  
2314

#### 2315 4.1.2.3 Matching 'name' attribute values

2316 For purposes of matching two 'name' attribute values for equality, such as in job validation (where a  
2317 client-supplied value for attribute "xxx" is checked to see if the value is among the values of the Printer  
2318 object's corresponding "xxx-supported" attribute), the following match rules apply:

- 2319       1. 'keyword' values never match 'name' values.
- 2320       2. 'name' (nameWithoutLanguage and nameWithLanguage) values match if (1) the name parts  
2321       match and (2) the Associated Natural-Language parts (see section 3.1.4.1) match. The matching  
2322       rules are:
  - 2323           a. the name parts match if the two names are identical character by character, except it is  
2324           RECOMMENDED that case be ignored. For example: 'Ajax-letter-head-white' MUST  
2325           match 'Ajax-letter-head-white' and SHOULD match 'ajax-letter-head-white' and 'AJAX-  
2326           LETTER-HEAD-WHITE'.
  - 2327           b. the Associated Natural-Language parts match if the shorter of the two meets the  
2328           syntactic requirements of RFC 1766 [RFC1766] and matches byte for byte with the  
2329           longer. For example, 'en' matches 'en', 'en-us' and 'en-gb', but matches neither 'fr' nor 'e'.

#### 2330 4.1.3 'keyword'

2331 The 'keyword' attribute syntax is a sequence of characters, length: 1 to 255, containing only the US-  
2332 ASCII [ASCII] encoded values for lowercase letters ("a" - "z"), digits ("0" - "9"), hyphen ("-"), dot ("."),  
2333 and underscore ("\_"). The first character MUST be a lowercase letter. Furthermore, keywords MUST  
2334 be in U.S. English.

2335 This syntax type is used for enumerating semantic identifiers of entities in the abstract protocol, i.e.,  
2336 entities identified in this document. Keywords are used as attribute names or values of attributes.  
2337 Unlike 'text' and 'name' attribute values, 'keyword' values MUST NOT use the Natural Language  
2338 Override mechanism, since they MUST always be US-ASCII and U.S. English.

2339 Keywords are for use in the protocol. A user interface will likely provide a mapping between protocol  
2340 keywords and displayable user-friendly words and phrases which are localized to the natural language of

2341 the user. While the keywords specified in this document MAY be displayed to users whose natural  
2342 language is U.S. English, they MAY be mapped to other U.S. English words for U.S. English users,  
2343 since the user interface is outside the scope of this document.

2344 In the definition for each attribute of this syntax type, the full set of defined keyword values for that  
2345 attribute are listed.

2346 When a keyword is used to represent an attribute (its name), it MUST be unique within the full scope of  
2347 all IPP objects and attributes. When a keyword is used to represent a value of an attribute, it MUST be  
2348 unique just within the scope of that attribute. That is, the same keyword MUST NOT be used for two  
2349 different values within the same attribute to mean two different semantic ideas. However, the same  
2350 keyword MAY be used across two or more attributes, representing different semantic ideas for each  
2351 attribute. Section 6.1 describes how the protocol can be extended with new keyword values. Examples  
2352 of attribute name keywords:

2353 "job-name"  
2354 "attributes-charset"  
2355

2356 Note: This document uses "type1", "type2", and "type3" prefixes to the "keyword" basic syntax to  
2357 indicate different levels of review for extensions (see section 6.1).

#### 2358 4.1.4 'enum'

2359 The 'enum' attribute syntax is an enumerated integer value that is in the range from 1 to 2\*\*31 - 1  
2360 (MAX). Each value has an associated 'keyword' name. In the definition for each attribute of this syntax  
2361 type, the full set of possible values for that attribute are listed. This syntax type is used for attributes for  
2362 which there are enum values assigned by other standards, such as SNMP MIBs. A number of attribute  
2363 enum values in this specification are also used for corresponding attributes in other standards  
2364 [RFC1759]. This syntax type is not used for attributes to which the administrator may assign values.  
2365 Section 6.1 describes how the protocol can be extended with new enum values.

2366 Enum values are for use in the protocol. A user interface will provide a mapping between protocol enum  
2367 values and displayable user-friendly words and phrases which are localized to the natural language of the  
2368 user. While the enum symbols specified in this document MAY be displayed to users whose natural  
2369 language is U.S. English, they MAY be mapped to other U.S. English words for U.S. English users,  
2370 since the user interface is outside the scope of this document.

2371 Note: SNMP MIBs use '2' for 'unknown' which corresponds to the IPP "out-of-band" value 'unknown'.  
2372 See the description of the "out-of-band" values at the beginning of Section 4.1. Therefore, attributes of  
2373 type 'enum' start at '3'.

2374 Note: This document uses "type1", "type2", and "type3" prefixes to the "enum" basic syntax to indicate  
2375 different levels of review for extensions (see section 6.1).

## 2376 4.1.5 'uri'

2377 The 'uri' attribute syntax is any valid Uniform Resource Identifier or URI [RFC2396]. Most often, URIs  
2378 are simply Uniform Resource Locators or URLs. The maximum length of URIs used as values of IPP  
2379 attributes is 1023 octets. Although most other IPP attribute syntax types allow for only lower-cased  
2380 values, this attribute syntax type conforms to the case-sensitive and case-insensitive rules specified in  
2381 [RFC2396]. See also [IPP-IIG] for a discussion of case in URIs.

## 2382 4.1.6 'uriScheme'

2383 The 'uriScheme' attribute syntax is a sequence of characters representing a URI scheme according to  
2384 RFC 2396 [RFC2396]. Though RFC 2396 requires that the values be case-insensitive, IPP requires all  
2385 lower case values in IPP attributes to simplify comparing by IPP clients and Printer objects. Standard  
2386 values for this syntax type are the following keywords:

2387 'http': for HTTP schemed URIs (e.g., "http:...")  
2388 'https': for use with HTTPS schemed URIs (e.g., "https:...") (not on IETF standards track)  
2389 'ftp': for FTP schemed URIs (e.g., "ftp:...")  
2390 'mailto': for SMTP schemed URIs (e.g., "mailto:...")  
2391 'file': for file schemed URIs (e.g., "file:...")  
2392

2393 A Printer object MAY support any URI 'scheme' that has been registered with IANA [IANA-MT]. The  
2394 maximum length of URI 'scheme' values used to represent IPP attribute values is 63 octets.

## 2395 4.1.7 'charset'

2396 The 'charset' attribute syntax is a standard identifier for a charset. A charset is a coded character set and  
2397 encoding scheme. Charsets are used for labeling certain document contents and 'text' and 'name'  
2398 attribute values. The syntax and semantics of this attribute syntax are specified in RFC 2046 [RFC2046]  
2399 and contained in the IANA character-set Registry [IANA-CS] according to the IANA procedures  
2400 [RFC2278]. Though RFC 2046 requires that the values be case-insensitive US-ASCII, IPP requires all  
2401 lower case values in IPP attributes to simplify comparing by IPP clients and Printer objects. When a  
2402 character-set in the IANA registry has more than one name (alias), the name labeled as "(preferred  
2403 MIME name)", if present, MUST be used.

2404 The maximum length of 'charset' values used to represent IPP attribute values is 63 octets.

2405 Some examples are:

2406 'utf-8': ISO 10646 Universal Multiple-Octet Coded Character Set (UCS) represented as the UTF-8  
2407 [RFC2279] transfer encoding scheme in which US-ASCII is a subset charset.  
2408 'us-ascii': 7-bit American Standard Code for Information Interchange (ASCII), ANSI X3.4-1986  
2409 [ASCII]. That standard defines US-ASCII, but RFC 2045 [RFC2045] eliminates most of the  
2410 control characters from conformant usage in MIME and IPP.

2411 'iso-8859-1': 8-bit One-Byte Coded Character Set, Latin Alphabet Nr 1 [ISO8859-1]. That standard  
2412 defines a coded character set that is used by Latin languages in the Western Hemisphere and  
2413 Western Europe. US-ASCII is a subset charset.

2414 'iso-10646-ucs-2': ISO 10646 Universal Multiple-Octet Coded Character Set (UCS) represented as  
2415 two octets (UCS-2), with the high order octet of each pair coming first (so-called Big Endian  
2416 integer).

2417

2418 Some attribute descriptions MAY place additional requirements on charset values that may be used, such  
2419 as REQUIRED values that MUST be supported or additional restrictions, such as requiring that the  
2420 charset have US-ASCII as a subset charset.

#### 2421 4.1.8 'naturalLanguage'

2422 The 'naturalLanguage' attribute syntax is a standard identifier for a natural language and optionally a  
2423 country. The values for this syntax type are defined by RFC 1766 [RFC1766]. Though RFC 1766  
2424 requires that the values be case-insensitive US-ASCII, IPP requires all lower case to simplify comparing  
2425 by IPP clients and Printer objects. Examples include:

2426 'en': for English

2427 'en-us': for US English

2428 'fr': for French

2429 'de': for German

2430

2431 The maximum length of 'naturalLanguage' values used to represent IPP attribute values is 63 octets.

#### 2432 4.1.9 'mimeType'

2433 The 'mimeType' attribute syntax is the Internet Media Type (sometimes called MIME type) as  
2434 defined by RFC 2046 [RFC2046] and registered according to the procedures of RFC 2048 [RFC2048]  
2435 for identifying a document format. The value MAY include a charset parameter, depending on the  
2436 specification of the Media Type in the IANA Registry [IANA-MT]. Although most other IPP syntax  
2437 types allow for only lower-cased values, this syntax type allows for mixed-case values which are case-  
2438 insensitive.

2439 Examples are:

2440 'text/html': An HTML document

2441 'text/plain': A plain text document in US-ASCII (RFC 2046 indicates that in the absence of the  
2442 charset parameter MUST mean US-ASCII rather than simply unspecified) [RFC2046].

2443 'text/plain; charset=US-ASCII': A plain text document in US-ASCII [52, 56].

2444 'text/plain; charset=ISO-8859-1': A plain text document in ISO 8859-1 (Latin 1) [ISO8859-1].

2445 'text/plain; charset=utf-8': A plain text document in ISO 10646 represented as UTF-8 [RFC2279]

2446 'application/postscript': A PostScript document [RFC2046]

2447 'application/vnd.hp-PCL': A PCL document [IANA-MT] (charset escape sequence embedded in the  
2448 document data)

2449 'application/pdf': Portable Document Format - see IANA MIME Media Type registry  
2450 'application/octet-stream': (REQUIRED) Auto-sense - see below

2451  
2452

2453 One special type is 'application/octet-stream'. If the Printer object supports this value, the Printer object  
2454 MUST be capable of auto-sensing the format of the document data. If the Printer object's default value  
2455 attribute "document-format-default" is set to 'application/octet-stream', the Printer object not only  
2456 supports auto-sensing of the document format, but will depend on the result of applying its auto-sensing  
2457 when the client does not supply the "document-format" attribute. If the client supplies a document  
2458 format value, the Printer MUST rely on the supplied attribute, rather than trust its auto-sensing  
2459 algorithm. To summarize:

- 2460 1. If the client does not supply a document format value, the Printer MUST rely on its default value  
2461 setting (which may be 'application/octet-stream' indicating an auto-sensing mechanism).
- 2462 2. If the client supplies a value other than 'application/octet-stream', the client is supplying valid  
2463 information about the format of the document data and the Printer object MUST trust the client  
2464 supplied value more than the outcome of applying an automatic format detection mechanism.  
2465 For example, the client may be requesting the printing of a PostScript file as a 'text/plain'  
2466 document. The Printer object MUST print a text representation of the PostScript commands  
2467 rather than interpret the stream of PostScript commands and print the result.
- 2468 3. If the client supplies a value of 'application/octet-stream', the client is indicating that the Printer  
2469 object MUST use its auto-sensing mechanism on the client supplied document data whether  
2470 auto-sensing is the Printer object's default or not.  
2471

2472 Note: Since the auto-sensing algorithm is probabilistic, if the client requests both auto-sensing  
2473 ("document-format" set to 'application/octet-stream') and true fidelity ("ipp-attribute-fidelity" set to  
2474 'true'), the Printer object might not be able to guarantee exactly what the end user intended (the auto-  
2475 sensing algorithm might mistake one document format for another ), but it is able to guarantee that its  
2476 auto-sensing mechanism be used.

2477 The maximum length of a 'mimeType' value to represent IPP attribute values is 255 octets.

#### 2478 4.1.10 'octetString'

2479 The 'octetString' attribute syntax is a sequence of octets encoded in a maximum of 1023 octets which is  
2480 indicated in sub-section headers using the notation: octetString(MAX). This syntax type is used for  
2481 opaque data.

#### 2482 4.1.11 'boolean'

2483 The 'boolean' attribute syntax has only two values: 'true' and 'false'.

## 2484 4.1.12 'integer'

2485 The 'integer' attribute syntax is an integer value that is in the range from  $-2^{31}$  (MIN) to  $2^{31} - 1$   
2486 (MAX). Each individual attribute may specify the range constraint explicitly in sub-section headers if  
2487 the range is different from the full range of possible integer values. For example: job-priority  
2488 (integer(1:100)) for the "job-priority" attribute. However, the enforcement of that additional constraint is  
2489 up to the IPP objects, not the protocol.

## 2490 4.1.13 'rangeOfInteger'

2491 The 'rangeOfInteger' attribute syntax is an ordered pair of integers that defines an inclusive range of  
2492 integer values. The first integer specifies the lower bound and the second specifies the upper bound. If a  
2493 range constraint is specified in the header description for an attribute in this document whose attribute  
2494 syntax is 'rangeOfInteger' (i.e., 'X:Y' indicating X as a minimum value and Y as a maximum value), then  
2495 the constraint applies to both integers.

## 2496 4.1.14 'dateTime'

2497 The 'dateTime' attribute syntax is a standard, fixed length, 11 octet representation of the "DateAndTime"  
2498 syntax as defined in RFC 1903 [RFC1903]. RFC 1903 also identifies an 8 octet representation of a  
2499 "DateAndTime" value, but IPP objects MUST use the 11 octet representation. A user interface will  
2500 provide a mapping between protocol dateTime values and displayable user-friendly words or  
2501 presentation values and phrases which are localized to the natural language and date format of the user.

## 2502 4.1.15 'resolution'

2503 The 'resolution' attribute syntax specifies a two-dimensional resolution in the indicated units. It consists  
2504 of 3 values: a cross feed direction resolution (positive integer value), a feed direction resolution (positive  
2505 integer value), and a units value. The semantics of these three components are taken from the Printer  
2506 MIB [RFC1759] suggested values. That is, the cross feed direction component resolution component is  
2507 the same as the prtMarkerAddressabilityXFeedDir object in the Printer MIB, the feed direction  
2508 component resolution component is the same as the prtMarkerAddressabilityFeedDir in the Printer MIB,  
2509 and the units component is the same as the prtMarkerAddressabilityUnit object in the Printer MIB  
2510 (namely, '3' indicates dots per inch and '4' indicates dots per centimeter). All three values MUST be  
2511 present even if the first two values are the same. Example: '300', '600', '3' indicates a 300 dpi cross-feed  
2512 direction resolution, a 600 dpi feed direction resolution, since a '3' indicates dots per inch (dpi).

## 2513 4.1.16 '1setOf X'

2514 The '1setOf X' attribute syntax is 1 or more values of attribute syntax type X. This syntax type is used  
2515 for multi-valued attributes. The syntax type is called '1setOf' rather than just 'setOf' as a reminder that  
2516 the set of values MUST NOT be empty (i.e., a set of size 0). Sets are normally unordered. However  
2517 each attribute description of this type may specify that the values MUST be in a certain order for that  
2518 attribute.

## 2519 4.2 Job Template Attributes

2520 Job Template attributes describe job processing behavior. Support for Job Template attributes by a  
2521 Printer object is OPTIONAL (see section 13.2.3 for a description of support for OPTIONAL attributes).  
2522 Also, clients OPTIONALLY supply Job Template attributes in create requests.

2523 Job Template attributes conform to the following rules. For each Job Template attribute called "xxx":

2524 1. If the Printer object supports "xxx" then it MUST support both a "xxx-default" attribute (unless  
2525 there is a "No" in the table below) and a "xxx-supported" attribute. If the Printer object doesn't  
2526 support "xxx", then it MUST support neither an "xxx-default" attribute nor an "xxx-supported"  
2527 attribute, and it MUST treat an attribute "xxx" supplied by a client as unsupported. An attribute  
2528 "xxx" may be supported for some document formats and not supported for other document  
2529 formats. For example, it is expected that a Printer object would only support "orientation-  
2530 requested" for some document formats (such as 'text/plain' or 'text/html') but not others (such as  
2531 'application/postscript').

2532  
2533 2. "xxx" is OPTIONALLY supplied by the client in a create request. If "xxx" is supplied, the client  
2534 is indicating a desired job processing behavior for this Job. When "xxx" is not supplied, the  
2535 client is indicating that the Printer object apply its default job processing behavior at job  
2536 processing time if the document content does not contain an embedded instruction indicating an  
2537 xxx-related behavior.

2538  
2539 Note: Since an administrator MAY change the default value attribute after a Job object has been  
2540 submitted but before it has been processed, the default value used by the Printer object at job  
2541 processing time may be different that the default value in effect at job submission time.

2542  
2543 3. The "xxx-supported" attribute is a Printer object attribute that describes which job processing  
2544 behaviors are supported by that Printer object. A client can query the Printer object to find out  
2545 what xxx-related behaviors are supported by inspecting the returned values of the "xxx-  
2546 supported" attribute.

2547  
2548 Note: The "xxx" in each "xxx-supported" attribute name is singular, even though an "xxx-  
2549 supported" attribute usually has more than one value, such as "job-sheet-supported", unless the  
2550 "xxx" Job Template attribute is plural, such as "finishings" or "sides". In such cases the "xxx-  
2551 supported" attribute names are: "finishings-supported" and "sides-supported".

2552  
2553 4. The "xxx-default" default value attribute describes what will be done at job processing time when  
2554 no other job processing information is supplied by the client (either explicitly as an IPP attribute  
2555 in the create request or implicitly as an embedded instruction within the document data).

2556  
2557 If an application wishes to present an end user with a list of supported values from which to choose, the  
2558 application SHOULD query the Printer object for its supported value attributes. The application  
2559 SHOULD also query the default value attributes. If the application then limits selectable values to only  
2560 those value that are supported, the application can guarantee that the values supplied by the client in the

2561 create request all fall within the set of supported values at the Printer. When querying the Printer, the  
2562 client MAY enumerate each attribute by name in the Get-Printer-Attributes Request, or the client MAY  
2563 just name the "job-template" group in order to get the complete set of supported attributes (both  
2564 supported and default attributes).

2565 The "finishings" attribute is an example of a Job Template attribute. It can take on a set of values such  
2566 as 'staple', 'punch', and/or 'cover'. A client can query the Printer object for the "finishings-supported"  
2567 attribute and the "finishings-default" attribute. The supported attribute contains a set of supported  
2568 values. The default value attribute contains the finishing value(s) that will be used for a new Job if the  
2569 client does not supply a "finishings" attribute in the create request and the document data does not  
2570 contain any corresponding finishing instructions. If the client does supply the "finishings" attribute in  
2571 the create request, the IPP object validates the value or values to make sure that they are a subset of the  
2572 supported values identified in the Printer object's "finishings-supported" attribute. See section 3.2.1.2.

2573 The table below summarizes the names and relationships for all Job Template attributes. The first  
2574 column of the table (labeled "Job Attribute") shows the name and syntax for each Job Template attribute  
2575 in the Job object. These are the attributes that can optionally be supplied by the client in a create request.  
2576 The last two columns (labeled "Printer: Default Value Attribute" and "Printer: Supported Values  
2577 Attribute") shows the name and syntax for each Job Template attribute in the Printer object (the default  
2578 value attribute and the supported values attribute). A "No" in the table means the Printer MUST NOT  
2579 support the attribute (that is, the attribute is simply not applicable). For brevity in the table, the 'text' and  
2580 'name' entries do not show the maximum length for each attribute.

2581	+	=====+	=====+	=====+
2582		Job Attribute	Printer: Default Value	Printer: Supported
2583			Attribute	Values Attribute
2584		=====+	=====+	=====+
2585		job-priority	job-priority-default	job-priority-supported
2586		(integer 1:100)	(integer 1:100)	(integer 1:100)
2587		-----+	-----+	-----+
2588		job-hold-until	job-hold-until-	job-hold-until-
2589		(type3 keyword	default	supported
2590		name)	(type3 keyword	(1setOf
2591			name)	type3 keyword   name)
2592		-----+	-----+	-----+
2593		job-sheets	job-sheets-default	job-sheets-supported
2594		(type3 keyword	(type3 keyword	(1setOf
2595		name)	name)	type3 keyword   name)
2596		-----+	-----+	-----+
2597		multiple-document-	multiple-document-	multiple-document-
2598		handling	handling-default	handling-supported
2599		(type2 keyword)	(type2 keyword)	(1setOf type2 keyword)
2600		-----+	-----+	-----+
2601		copies	copies-default	copies-supported
2602		(integer (1:MAX))	(integer (1:MAX))	(rangeOfInteger
2603				(1:MAX))
2604		-----+	-----+	-----+
2605		finishings	finishings-default	finishings-supported
2606		(1setOf type2 enum)	(1setOf type2 enum)	(1setOf type2 enum)
2607		-----+	-----+	-----+
2608		page-ranges	No	page-ranges-
2609		(1setOf		supported (boolean)
2610		rangeOfInteger		
2611		(1:MAX))		
2612		-----+	-----+	-----+
2613		sides	sides-default	sides-supported
2614		(type2 keyword)	(type2 keyword)	(1setOf type2 keyword)
2615		-----+	-----+	-----+
2616		number-up	number-up-default	number-up-supported
2617		(integer (1:MAX))	(integer (1:MAX))	(1setOf integer
2618				(1:MAX)
2619				rangeOfInteger
2620				(1:MAX))
2621		-----+	-----+	-----+
2622		orientation-	orientation-requested-	orientation-requested-
2623		requested	default	supported
2624		(type2 enum)	(type2 enum)	(1setOf type2 enum)
2625		-----+	-----+	-----+
2626		media	media-default	media-supported
2627		(type3 keyword	(type3 keyword	(1setOf
2628		name)	name)	type3 keyword   name)
2629				
2630				media-ready

2631			(1setOf
2632			type3 keyword   name)
2633	+-----+-----+-----+		
2634	printer-resolution	printer-resolution-	printer-resolution-
2635	(resolution)	default	supported
2636		(resolution)	(1setOf resolution)
2637	+-----+-----+-----+		
2638	print-quality	print-quality-default	print-quality-
2639	(type2 enum)	(type2 enum)	supported
2640			(1setOf type2 enum)
2641	+-----+-----+-----+		

2642

2643

#### 2644 4.2.1 job-priority (integer(1:100))

2645 This attribute specifies a priority for scheduling the Job. A higher value specifies a higher priority. The  
 2646 value 1 indicates the lowest possible priority. The value 100 indicates the highest possible priority.  
 2647 Among those jobs that are ready to print, a Printer MUST print all jobs with a priority value of n before  
 2648 printing those with a priority value of n-1 for all n.

2649 If the Printer object supports this attribute, it MUST always support the full range from 1 to 100. No  
 2650 administrative restrictions are permitted. This way an end-user can always make full use of the entire  
 2651 range with any Printer object. If privileged jobs are implemented outside IPP/1.1, they MUST have  
 2652 priorities higher than 100, rather than restricting the range available to end-users.

2653 If the client does not supply this attribute and this attribute is supported by the Printer object, the Printer  
 2654 object MUST use the value of the Printer object's "job-priority-default" at job submission time (unlike  
 2655 most Job Template attributes that are used if necessary at job processing time).

2656 The syntax for the "job-priority-supported" is also integer(1:100). This single integer value indicates the  
 2657 number of priority levels supported. The Printer object MUST take the value supplied by the client and  
 2658 map it to the closest integer in a sequence of n integers values that are evenly distributed over the range  
 2659 from 1 to 100 using the formula:

2660 
$$\text{roundToNearestInt}((100x+50)/n)$$

2661 where n is the value of "job-priority-supported" and x ranges from 0 through n-1.

2662 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,  
 2663 the sequence of values is: 17, 50 and 83; if n = 10, the sequence of values is: 5, 15, 25, 35, 45, 55, 65,  
 2664 75, 85, and 95; if n = 100, the sequence of values is: 1, 2, 3, ... 100.

2665 If the value of the Printer object's "job-priority-supported" is 10 and the client supplies values in the  
 2666 range 1 to 10, the Printer object maps them to 5, in the range 11 to 20, the Printer object maps them to  
 2667 15, etc.

2668 4.2.2 job-hold-until (type3 keyword | name (MAX))

2669 This attribute specifies the named time period during which the Job MUST become a candidate for  
2670 printing.

2671 Standard keyword values for named time periods are:

- 2672     'no-hold': immediately, if there are not other reasons to hold the job
  - 2673     'indefinite': - the job is held indefinitely, until a client performs a Release-Job (section 3.3.6)
  - 2674     'day-time': during the day
  - 2675     'evening': evening
  - 2676     'night': night
  - 2677     'weekend': weekend
  - 2678     'second-shift': second-shift (after close of business)
  - 2679     'third-shift': third-shift (after midnight)
- 2680

2681 An administrator MUST associate allowable print times with a named time period (by means outside the  
2682 scope of this IPP/1.1 document). An administrator is encouraged to pick names that suggest the type of  
2683 time period. An administrator MAY define additional values using the 'name' or 'keyword' attribute  
2684 syntax, depending on implementation.

2685 If the value of this attribute specifies a time period that is in the future, the Printer MUST add the 'job-  
2686 hold-until-specified' value to the job's "job-state-reasons" attribute, move the job to the 'pending-held'  
2687 state, and MUST NOT schedule the job for printing until the specified time-period arrives. When the  
2688 specified time period arrives, the Printer MUST remove the 'job-hold-until-specified' value from the  
2689 job's "job-state-reason" attribute and, if there are no other job state reasons that keep the job in the  
2690 'pending-held' state, the Printer MUST consider the job as a candidate for processing by moving the job  
2691 to the 'pending' state.

2692 If this job attribute value is the named value 'no-hold', or the specified time period has already started,  
2693 the job MUST be a candidate for processing immediately.

2694 If the client does not supply this attribute and this attribute is supported by the Printer object, the Printer  
2695 object MUST use the value of the Printer object's "job-hold-until-default" at job submission time (unlike  
2696 most Job Template attributes that are used if necessary at job processing time).

2697 4.2.3 job-sheets (type3 keyword | name(MAX))

2698 This attribute determines which job start/end sheet(s), if any, MUST be printed with a job.

2699 Standard keyword values are:

- 2700     'none': no job sheet is printed
  - 2701     'standard': one or more site specific standard job sheets are printed, e.g. a single start sheet or both  
2702         start and end sheet is printed
- 2703

2704 An administrator MAY define additional values using the 'name' or 'keyword' attribute syntax, depending  
2705 on implementation.

2706 Note: The effect of this attribute on jobs with multiple documents MAY be affected by the "multiple-  
2707 document-handling" job attribute (section 4.2.4), depending on the job sheet semantics.

#### 2708 4.2.4 multiple-document-handling (type2 keyword)

2709 This attribute is relevant only if a job consists of two or more documents. The attribute controls finishing  
2710 operations and the placement of one or more print-stream pages into impressions and onto media sheets.  
2711 When the value of the "copies" attribute exceeds 1, it also controls the order in which the copies that  
2712 result from processing the documents are produced. For the purposes of this explanations, if "a"  
2713 represents an instance of document data, then the result of processing the data in document "a" is a  
2714 sequence of media sheets represented by "a(\*)".

2715 Standard keyword values are:

2716 'single-document': If a Job object has multiple documents, say, the document data is called a and b,  
2717 then the result of processing all the document data (a and then b) MUST be treated as a single  
2718 sequence of media sheets for finishing operations; that is, finishing would be performed on the  
2719 concatenation of the sequences a(\*),b(\*). The Printer object MUST NOT force the data in each  
2720 document instance to be formatted onto a new print-stream page, nor to start a new impression  
2721 on a new media sheet. If more than one copy is made, the ordering of the sets of media sheets  
2722 resulting from processing the document data MUST be a(\*), b(\*), a(\*), b(\*), ..., and the Printer  
2723 object MUST force each copy (a(\*),b(\*)) to start on a new media sheet.

2724 'separate-documents-uncollated-copies': If a Job object has multiple documents, say, the document  
2725 data is called a and b, then the result of processing the data in each document instance MUST be  
2726 treated as a single sequence of media sheets for finishing operations; that is, the sets a(\*) and b(\*)  
2727 would each be finished separately. The Printer object MUST force each copy of the result of  
2728 processing the data in a single document to start on a new media sheet. If more than one copy is  
2729 made, the ordering of the sets of media sheets resulting from processing the document data  
2730 MUST be a(\*), a(\*), ..., b(\*), b(\*) ... .

2731 'separate-documents-collated-copies': If a Job object has multiple documents, say, the document data  
2732 is called a and b, then the result of processing the data in each document instance MUST be  
2733 treated as a single sequence of media sheets for finishing operations; that is, the sets a(\*) and b(\*)  
2734 would each be finished separately. The Printer object MUST force each copy of the result of  
2735 processing the data in a single document to start on a new media sheet. If more than one copy is  
2736 made, the ordering of the sets of media sheets resulting from processing the document data  
2737 MUST be a(\*), b(\*), a(\*), b(\*), ... .

2738 'single-document-new-sheet': Same as 'single-document', except that the Printer object MUST  
2739 ensure that the first impression of each document instance in the job is placed on a new media  
2740 sheet. This value allows multiple documents to be stapled together with a single staple where  
2741 each document starts on a new sheet.

2742

2743 The 'single-document' value is the same as 'separate-documents-collated-copies' with respect to ordering  
2744 of print-stream pages, but not media sheet generation, since 'single-document' will put the first page of

2745 the next document on the back side of a sheet if an odd number of pages have been produced so far for  
 2746 the job, while 'separate-documents-collated-copies' always forces the next document or document copy  
 2747 on to a new sheet. In addition, if the "finishings" attribute specifies 'staple', then with 'single-document',  
 2748 documents a and b are stapled together as a single document with no regard to new sheets, with 'single-  
 2749 document-new-sheet', documents a and b are stapled together as a single document, but document b  
 2750 starts on a new sheet, but with 'separate-documents-uncollated-copies' and 'separate-documents-collated-  
 2751 copies', documents a and b are stapled separately.

2752 Note: None of these values provide means to produce uncollated sheets within a document, i.e., where  
 2753 multiple copies of sheet n are produced before sheet n+1 of the same document.

2754 The relationship of this attribute and the other attributes that control document processing is described in  
 2755 section 16.3.

#### 2756 4.2.5 copies (integer(1:MAX))

2757 This attribute specifies the number of copies to be printed.

2758 On many devices the supported number of collated copies will be limited by the number of physical  
 2759 output bins on the device, and may be different from the number of uncollated copies which can be  
 2760 supported.

2761 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-  
 2762 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other  
 2763 attributes that control document processing is described in section 16.3.

#### 2764 4.2.6 finishings (1setOf type2 enum)

2765 This attribute identifies the finishing operations that the Printer uses for each copy of each printed  
 2766 document in the Job. For Jobs with multiple documents, the "multiple-document-handling" attribute  
 2767 determines what constitutes a "copy" for purposes of finishing.

2768 Standard enum values are:

2769	Value	Symbolic Name and Description
2770		
2771	'3'	'none': Perform no finishing
2772	'4'	'staple': Bind the document(s) with one or more staples. The exact number and placement 2773 of the staples is site-defined.
2774	'5'	'punch': This value indicates that holes are required in the finished document. The exact 2775 number and placement of the holes is site-defined The punch specification MAY 2776 be satisfied (in a site- and implementation-specific manner) either by 2777 drilling/punching, or by substituting pre-drilled media.
2778	'6'	'cover': This value is specified when it is desired to select a non-printed (or pre-printed) 2779 cover for the document. This does not supplant the specification of a printed cover 2780 (on cover stock medium) by the document itself.

- 2781 7' 'bind': This value indicates that a binding is to be applied to the document; the type and  
2782 placement of the binding is site-defined.  
2783
- 2784 8' 'saddle-stitch': Bind the document(s) with one or more staples (wire stitches) along the  
2785 middle fold. The exact number and placement of the staples and the middle fold  
2786 is implementation and/or site-defined.
- 2787 9' 'edge-stitch': Bind the document(s) with one or more staples (wire stitches) along one  
2788 edge. The exact number and placement of the staples is implementation and/or  
2789 site-defined.
- 2790 '10'-'19' reserved for future generic finishing enum values.
- 2791 The following values are more specific; they indicate a corner or an edge as if the document were a  
2792 portrait document (see below):
- 2793 20' 'staple-top-left': Bind the document(s) with one or more staples in the top left corner.  
2794 21' 'staple-bottom-left': Bind the document(s) with one or more staples in the bottom left  
2795 corner.  
2796 22' 'staple-top-right': Bind the document(s) with one or more staples in the top right corner.  
2797 23' 'staple-bottom-right': Bind the document(s) with one or more staples in the bottom right  
2798 corner.  
2799 24' 'edge-stitch-left': Bind the document(s) with one or more staples (wire stitches) along the  
2800 left edge. The exact number and placement of the staples is implementation  
2801 and/or site-defined.  
2802 25' 'edge-stitch-top': Bind the document(s) with one or more staples (wire stitches) along the  
2803 top edge. The exact number and placement of the staples is implementation  
2804 and/or site-defined.  
2805 26' 'edge-stitch-right': Bind the document(s) with one or more staples (wire stitches) along  
2806 the right edge. The exact number and placement of the staples is implementation  
2807 and/or site-defined.  
2808 27' 'edge-stitch-bottom': Bind the document(s) with one or more staples (wire stitches) along  
2809 the bottom edge. The exact number and placement of the staples is  
2810 implementation and/or site-defined.  
2811 28' 'staple-dual-left': Bind the document(s) with two staples (wire stitches) along the left  
2812 edge assuming a portrait document (see above).  
2813 29' 'staple-dual-top': Bind the document(s) with two staples (wire stitches) along the top  
2814 edge assuming a portrait document (see above).  
2815 30' 'staple-dual-right': Bind the document(s) with two staples (wire stitches) along the right  
2816 edge assuming a portrait document (see above).  
2817 31' 'staple-dual-bottom': Bind the document(s) with two staples (wire stitches) along the  
2818 bottom edge assuming a portrait document (see above).

2819 The 'staple-xxx' values are specified with respect to the document as if the document were a portrait  
2820 document. If the document is actually a landscape or a reverse-landscape document, the client supplies  
2821 the appropriate transformed value. For example, to position a staple in the upper left hand corner of a  
2822 landscape document when held for reading, the client supplies the 'staple-bottom-left' value (since  
2823 landscape is defined as a +90 degree rotation from portrait, i.e., anti-clockwise). On the other hand, to  
2824 position a staple in the upper left hand corner of a reverse-landscape document when held for reading,

2825 the client supplies the 'staple-top-right' value (since reverse-landscape is defined as a -90 degree rotation  
2826 from portrait, i.e., clockwise).

2827 The angle (vertical, horizontal, angled) of each staple with respect to the document depends on the  
2828 implementation which may in turn depend on the value of the attribute.

2829 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-  
2830 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other  
2831 attributes that control document processing is described in section 16.3.

2832 If the client supplies a value of 'none' along with any other combination of values, it is the same as if  
2833 only that other combination of values had been supplied (that is the 'none' value has no effect).

#### 2834 4.2.7 page-ranges (1setOf rangeOfInteger (1:MAX))

2835 This attribute identifies the range(s) of print-stream pages that the Printer object uses for each copy of  
2836 each document which are to be printed. Nothing is printed for any pages identified that do not exist in  
2837 the document(s). Ranges MUST be in ascending order, for example: 1-3, 5-7, 15-19 and MUST NOT  
2838 overlap, so that a non-spooling Printer object can process the job in a single pass. If the ranges are not  
2839 ascending or are overlapping, the IPP object MUST reject the request and return the 'client-error-bad-  
2840 request' status code. The attribute is associated with print-stream pages not application-numbered pages  
2841 (for example, the page numbers found in the headers and or footers for certain word processing  
2842 applications).

2843 For Jobs with multiple documents, the "multiple-document-handling" attribute determines what  
2844 constitutes a "copy" for purposes of the specified page range(s). When "multiple-document-handling" is  
2845 'single-document', the Printer object MUST apply each supplied page range once to the concatenation of  
2846 the print-stream pages. For example, if there are 8 documents of 10 pages each, the page-range '4:60'  
2847 prints the pages in the 5th and 6th documents as a single document and none of the pages of the other  
2848 documents are printed. When "multiple-document-handling" is 'separate-document-uncollated-copies' or  
2849 'separate-document-collated-copies', the Printer object MUST apply each supplied page range repeatedly  
2850 to each document copy. For the same job, the page-range '1:3, 10:10' would print the first 3 pages and  
2851 the 10th page of each of the 8 documents in the Job, as 8 separate documents.

2852 In most cases, the exact pages to be printed will be generated by a device driver and this attribute would  
2853 not be required. However, when printing an archived document which has already been formatted, the  
2854 end user may elect to print just a subset of the pages contained in the document. In this case, if page-  
2855 range = n.m is specified, the first page to be printed will be page n. All subsequent pages of the  
2856 document will be printed through and including page m.

2857 "page-ranges-supported" is a boolean value indicating whether or not the printer is capable of supporting  
2858 the printing of page ranges. This capability may differ from one PDL to another. There is no "page-  
2859 ranges-default" attribute. If the "page-ranges" attribute is not supplied by the client, all pages of the  
2860 document will be printed.

2861 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-  
 2862 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other  
 2863 attributes that control document processing is described in section 16.3.

#### 2864 4.2.8 sides (type2 keyword)

2865 This attribute specifies how print-stream pages are to be imposed upon the sides of an instance of a  
 2866 selected medium, i.e., an impression.

2867 The standard keyword values are:

2868 'one-sided': imposes each consecutive print-stream page upon the same side of consecutive media  
 2869 sheets.

2870 'two-sided-long-edge': imposes each consecutive pair of print-stream pages upon front and back sides  
 2871 of consecutive media sheets, such that the orientation of each pair of print-stream pages on the  
 2872 medium would be correct for the reader as if for binding on the long edge. This imposition is  
 2873 sometimes called 'duplex' or 'head-to-head'.

2874 'two-sided-short-edge': imposes each consecutive pair of print-stream pages upon front and back  
 2875 sides of consecutive media sheets, such that the orientation of each pair of print-stream pages on  
 2876 the medium would be correct for the reader as if for binding on the short edge. This imposition  
 2877 is sometimes called 'tumble' or 'head-to-toe'.  
 2878

2879 'two-sided-long-edge', 'two-sided-short-edge', 'tumble', and 'duplex' all work the same for portrait or  
 2880 landscape. However 'head-to-toe' is 'tumble' in portrait but 'duplex' in landscape. 'head-to-head' also  
 2881 switches between 'duplex' and 'tumble' when using portrait and landscape modes.

2882 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-  
 2883 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other  
 2884 attributes that control document processing is described in section 16.3.

#### 2885 4.2.9 number-up (integer(1:MAX))

2886 This attribute specifies the number of print-stream pages to impose upon a single side of an instance of a  
 2887 selected medium. For example, if the value is:

2888 Value	Description
2889 '1'	the Printer MUST place one print-stream page on a single side of an instance of the 2890 selected medium (MAY add some sort of translation, scaling, or rotation).
2891 '2'	the Printer MUST place two print-stream pages on a single side of an instance of the 2892 selected medium (MAY add some sort of translation, scaling, or rotation).
2893 '4'	the Printer MUST place four print-stream pages on a single side of an instance of the 2894 selected medium (MAY add some sort of translation, scaling, or rotation). 2895 2896

2897 This attribute primarily controls the translation, scaling and rotation of print-stream pages.

2898 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-  
2899 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other  
2900 attributes that control document processing is described in section 16.3.

#### 2901 4.2.10 orientation-requested (type2 enum)

2902 This attribute indicates the desired orientation for printed print-stream pages; it does not describe the  
2903 orientation of the client-supplied print-stream pages.

2904 For some document formats (such as 'application/postscript'), the desired orientation of the print-stream  
2905 pages is specified within the document data. This information is generated by a device driver prior to  
2906 the submission of the print job. Other document formats (such as 'text/plain') do not include the notion  
2907 of desired orientation within the document data. In the latter case it is possible for the Printer object to  
2908 bind the desired orientation to the document data after it has been submitted. It is expected that a Printer  
2909 object would only support "orientations-requested" for some document formats (e.g., 'text/plain' or  
2910 'text/html') but not others (e.g., 'application/postscript'). This is no different than any other Job Template  
2911 attribute since section 4.2, item 1, points out that a Printer object may support or not support any Job  
2912 Template attribute based on the document format supplied by the client. However, a special mention is  
2913 made here since it is very likely that a Printer object will support "orientation-requested" for only a  
2914 subset of the supported document formats.

2915 Standard enum values are:

2916	Value	Symbolic Name and Description
2917		
2918	'3'	'portrait': The content will be imaged across the short edge of the medium.
2919	'4'	'landscape': The content will be imaged across the long edge of the medium. Landscape
2920		is defined to be a rotation of the print-stream page to be imaged by +90 degrees
2921		with respect to the medium (i.e. anti-clockwise) from the portrait orientation.
2922		Note: The +90 direction was chosen because simple finishing on the long edge is
2923		the same edge whether portrait or landscape
2924	'5'	'reverse-landscape': The content will be imaged across the long edge of the medium.
2925		Reverse-landscape is defined to be a rotation of the print-stream page to be
2926		imaged by -90 degrees with respect to the medium (i.e. clockwise) from the
2927		portrait orientation. Note: The 'reverse-landscape' value was added because some
2928		applications rotate landscape -90 degrees from portrait, rather than +90 degrees.
2929	'6'	'reverse-portrait': The content will be imaged across the short edge of the medium.
2930		Reverse-portrait is defined to be a rotation of the print-stream page to be imaged
2931		by 180 degrees with respect to the medium from the portrait orientation. Note:
2932		The 'reverse-portrait' value was added for use with the "finishings" attribute in
2933		cases where the opposite edge is desired for finishing a portrait document on
2934		simple finishing devices that have only one finishing position. Thus a 'text/plain'
2935		portrait document can be stapled "on the right" by a simple finishing device as is
2936		common use with some middle eastern languages such as Hebrew.
2937		

2938 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-  
2939 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other  
2940 attributes that control document processing is described in section 16.3.

#### 2941 4.2.11 media (type3 keyword | name(MAX))

2942 This attribute identifies the medium that the Printer uses for all impressions of the Job.

2943 The values for "media" include medium-names, medium-sizes, input-trays and electronic forms so that  
2944 one attribute specifies the media. If a Printer object supports a medium name as a value of this attribute,  
2945 such a medium name implicitly selects an input-tray that contains the specified medium. If a Printer  
2946 object supports a medium size as a value of this attribute, such a medium size implicitly selects a  
2947 medium name that in turn implicitly selects an input-tray that contains the medium with the specified  
2948 size. If a Printer object supports an input-tray as the value of this attribute, such an input-tray implicitly  
2949 selects the medium that is in that input-tray at the time the job prints. This case includes manual-feed  
2950 input-trays. If a Printer object supports an electronic form as the value of this attribute, such an  
2951 electronic form implicitly selects a medium-name that in turn implicitly selects an input-tray that  
2952 contains the medium specified by the electronic form. The electronic form also implicitly selects an  
2953 image that the Printer MUST merge with the document data as its prints each page.

2954 Standard keyword values are (taken from ISO DPA and the Printer MIB) and are listed in section 15. An  
2955 administrator MAY define additional values using the 'name' or 'keyword' attribute syntax, depending on  
2956 implementation.

2957 There is also an additional Printer attribute named "media-ready" which differs from "media-supported"  
2958 in that legal values only include the subset of "media-supported" values that are physically loaded and  
2959 ready for printing with no operator intervention required. If an IPP object supports "media-supported", it  
2960 NEED NOT support "media-ready".

2961 The relationship of this attribute and the other attributes that control document processing is described in  
2962 section 16.3.

#### 2963 4.2.12 printer-resolution (resolution)

2964 This attribute identifies the resolution that Printer uses for the Job.

#### 2965 4.2.13 print-quality (type2 enum)

2966 This attribute specifies the print quality that the Printer uses for the Job.

2967 The standard enum values are:

2968	Value	Symbolic Name and Description
2969		
2970	'3'	'draft': lowest quality available on the printer
2971	'4'	'normal': normal or intermediate quality on the printer

2972        '5'        'high': highest quality available on the printer  
2973

#### 2974 4.3 Job Description Attributes

2975 The attributes in this section form the attribute group called "job-description". The following table  
2976 summarizes these attributes. The third column indicates whether the attribute is a REQUIRED attribute  
2977 that MUST be supported by Printer objects. If it is not indicated as REQUIRED, then it is OPTIONAL.  
2978 The maximum size in octets for 'text' and 'name' attributes is indicated in parentheses.

	Attribute	Syntax	REQUIRED?
2979			
2980			
2981			
2982	job-uri	uri	REQUIRED
2983			
2984	job-id	integer(1:MAX)	REQUIRED
2985			
2986	job-printer-uri	uri	REQUIRED
2987			
2988	job-more-info	uri	
2989			
2990	job-name	name (MAX)	REQUIRED
2991			
2992	job-originating-user-name	name (MAX)	REQUIRED
2993			
2994	job-state	type1 enum	REQUIRED
2995			
2996	job-state-reasons	1setOf type2 keyword	
2997			
2998	job-state-message	text (MAX)	
2999			
3000	number-of-documents	integer (0:MAX)	
3001			
3002	output-device-assigned	name (127)	
3003			
3004	time-at-creation	integer (0:MAX)	
3005			
3006	time-at-processing	integer (0:MAX)	
3007			
3008	time-at-completed	integer (0:MAX)	
3009			
3010	number-of-intervening-jobs	integer (0:MAX)	
3011			
3012	job-message-from-operator	text (127)	
3013			
3014	job-k-octets	integer (0:MAX)	
3015			
3016	job-impressions	integer (0:MAX)	
3017			
3018	job-media-sheets	integer (0:MAX)	
3019			
3020	job-k-octets-processed	integer (0:MAX)	
3021			
3022	job-impressions-completed	integer (0:MAX)	
3023			
3024	job-media-sheets-completed	integer (0:MAX)	
3025			
3026	attributes-charset	charset	REQUIRED
3027			
3028	attributes-natural-language	naturalLanguage	REQUIRED

3029 +-----+-----+-----+-----+  
3030  
3031

#### 3032 4.3.1 job-uri (uri)

3033 This REQUIRED attribute contains the URI for the job. The Printer object, on receipt of a new job,  
3034 generates a URI which identifies the new Job. The Printer object returns the value of the "job-uri"  
3035 attribute as part of the response to a create request. The precise format of a Job URI is implementation  
3036 dependent. If the Printer object supports more than one URI and there is some relationship between the  
3037 newly formed Job URI and the Printer object's URI, the Printer object uses the Printer URI supplied by  
3038 the client in the create request. For example, if the create request comes in over a secure channel, the  
3039 new Job URI MUST use the same secure channel. This can be guaranteed because the Printer object is  
3040 responsible for generating the Job URI and the Printer object is aware of its security configuration and  
3041 policy as well as the Printer URI used in the create request.

3042 For a description of this attribute and its relationship to "job-id" and "job-printer-uri" attribute, see the  
3043 discussion in section 2.4 on "Object Identity".

#### 3044 4.3.2 job-id (integer(1:MAX))

3045 This REQUIRED attribute contains the ID of the job. The Printer, on receipt of a new job, generates an  
3046 ID which identifies the new Job on that Printer. The Printer returns the value of the "job-id" attribute as  
3047 part of the response to a create request. The 0 value is not included to allow for compatibility with  
3048 SNMP index values which also cannot be 0.

3049 For a description of this attribute and its relationship to "job-uri" and "job-printer-uri" attribute, see the  
3050 discussion in section 2.4 on "Object Identity".

#### 3051 4.3.3 job-printer-uri (uri)

3052 This REQUIRED attribute identifies the Printer object that created this Job object. When a Printer  
3053 object creates a Job object, it populates this attribute with the Printer object URI that was used in the  
3054 create request. This attribute permits a client to identify the Printer object that created this Job object  
3055 when only the Job object's URI is available to the client. The client queries the creating Printer object to  
3056 determine which languages, charsets, operations, are supported for this Job.

3057 For a description of this attribute and its relationship to "job-uri" and "job-id" attribute, see the  
3058 discussion in section 2.4 on "Object Identity".

#### 3059 4.3.4 job-more-info (uri)

3060 Similar to "printer-more-info", this attribute contains the URI referencing some resource with more  
3061 information about this Job object, perhaps an HTML page containing information about the Job.

## 3062 4.3.5 job-name (name(MAX))

3063 This REQUIRED attribute is the name of the job. It is a name that is more user friendly than the "job-  
 3064 uri" attribute value. It does not need to be unique between Jobs. The Job's "job-name" attribute is set to  
 3065 the value supplied by the client in the "job-name" operation attribute in the create request (see Section  
 3066 3.2.1.1). If, however, the "job-name" operation attribute is not supplied by the client in the create  
 3067 request, the Printer object, on creation of the Job, MUST generate a name. The printer SHOULD  
 3068 generate the value of the Job's "job-name" attribute from the first of the following sources that produces  
 3069 a value: 1) the "document-name" operation attribute of the first (or only) document, 2) the "document-  
 3070 URI" attribute of the first (or only) document, or 3) any other piece of Job specific and/or Document  
 3071 Content information.

## 3072 4.3.6 job-originating-user-name (name(MAX))

3073 This REQUIRED attribute contains the name of the end user that submitted the print job. The Printer  
 3074 object sets this attribute to the most authenticated printable name that it can obtain from the  
 3075 authentication service over which the IPP operation was received. Only if such is not available, does the  
 3076 Printer object use the value supplied by the client in the "requesting-user-name" operation attribute of the  
 3077 create operation (see Section 8).

3078 Note: The Printer object needs to keep an internal originating user id of some form, typically as a  
 3079 credential of a principal, with the Job object. Since such an internal attribute is implementation-  
 3080 dependent and not of interest to clients, it is not specified as a Job Description attribute. This originating  
 3081 user id is used for authorization checks (if any) on all subsequent operation.

## 3082 4.3.7 job-state (type1 enum)

3083 This REQUIRED attribute identifies the current state of the job. Even though the IPP protocol defines  
 3084 seven values for job states (plus the out-of-band 'unknown' value - see Section 4.1), implementations  
 3085 only need to support those states which are appropriate for the particular implementation. In other  
 3086 words, a Printer supports only those job states implemented by the output device and available to the  
 3087 Printer object implementation.

3088 Standard enum values are:

3089	Values	Symbolic Name and Description
3090		
3091	'3'	'pending': The job is a candidate to start processing, but is not yet processing.
3092		
3093	'4'	'pending-held': The job is not a candidate for processing for any number of reasons but 3094 will return to the 'pending' state as soon as the reasons are no longer present. The 3095 job's "job-state-reason" attribute MUST indicate why the job is no longer a 3096 candidate for processing.
3097		
3098	'5'	'processing': One or more of:
3099		

- 3100 1. the job is using, or is attempting to use, one or more purely software processes  
3101 that are analyzing, creating, or interpreting a PDL, etc.,  
3102 2. the job is using, or is attempting to use, one or more hardware devices that are  
3103 interpreting a PDL, making marks on a medium, and/or performing finishing,  
3104 such as stapling, etc.,  
3105 3. the Printer object has made the job ready for printing, but the output device is  
3106 not yet printing it, either because the job hasn't reached the output device or  
3107 because the job is queued in the output device or some other spooler, awaiting the  
3108 output device to print it.  
3109

3110 When the job is in the 'processing' state, the entire job state includes the detailed  
3111 status represented in the printer's "printer-state", "printer-state-reasons", and  
3112 "printer-state-message" attributes.

3113 Implementations MAY, though they NEED NOT, include additional values in the  
3114 job's "job-state-reasons" attribute to indicate the progress of the job, such as  
3115 adding the 'job-printing' value to indicate when the output device is actually  
3116 making marks on paper and/or the 'processing-to-stop-point' value to indicate that  
3117 the IPP object is in the process of canceling or aborting the job. Most  
3118 implementations won't bother with this nuance.

3119  
3120 '6' 'processing-stopped': The job has stopped while processing for any number of reasons  
3121 and will return to the 'processing' state as soon as the reasons are no longer  
3122 present.  
3123

3124 The job's "job-state-reason" attribute MAY indicate why the job has stopped  
3125 processing. For example, if the output device is stopped, the 'printer-stopped'  
3126 value MAY be included in the job's "job-state-reasons" attribute.

3127  
3128 Note: When an output device is stopped, the device usually indicates its condition  
3129 in human readable form locally at the device. A client can obtain more complete  
3130 device status remotely by querying the Printer object's "printer-state", "printer-  
3131 state-reasons" and "printer-state-message" attributes.  
3132

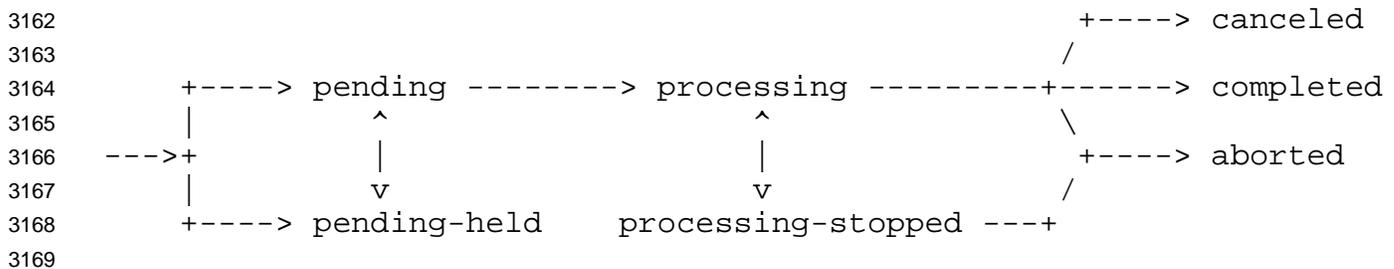
3133 '7' 'canceled': The job has been canceled by a Cancel-Job operation and the Printer object  
3134 has completed canceling the job and all job status attributes have reached their  
3135 final values for the job. While the Printer object is canceling the job, the job  
3136 remains in its current state, but the job's "job-state-reasons" attribute SHOULD  
3137 contain the 'processing-to-stop-point' value and one of the 'canceled-by-user',  
3138 'canceled-by-operator', or 'canceled-at-device' value. When the job moves to the  
3139 'canceled' state, the 'processing-to-stop-point' value, if present, MUST be  
3140 removed, but the 'canceled-by-xxx', if present, MUST remain.  
3141

3142 '8' 'aborted': The job has been aborted by the system, usually while the job was in the  
 3143 'processing' or 'processing-stopped' state and the Printer has completed aborting  
 3144 the job and all job status attributes have reached their final values for the job.  
 3145 While the Printer object is aborting the job, the job remains in its current state, but  
 3146 the job's "job-state-reasons" attribute SHOULD contain the 'processing-to-stop-  
 3147 point' and 'aborted-by-system' values. When the job moves to the 'aborted' state,  
 3148 the 'processing-to-stop-point' value, if present, MUST be removed, but the  
 3149 'aborted-by-system' value, if present, MUST remain.

3151 '9' 'completed': The job has completed successfully or with warnings or errors after  
 3152 processing and all of the job media sheets have been successfully stacked in the  
 3153 appropriate output bin(s) and all job status attributes have reached their final  
 3154 values for the job. The job's "job-state-reasons" attribute SHOULD contain one  
 3155 of: 'completed-successfully', 'completed-with-warnings', or 'completed-with-errors'  
 3156 values.  
 3157

3158 The final value for this attribute MUST be one of: 'completed', 'canceled', or 'aborted' before the Printer  
 3159 removes the job altogether. The length of time that jobs remain in the 'canceled', 'aborted', and  
 3160 'completed' states depends on implementation. See section 4.3.7.1.

3161 The following figure shows the normal job state transitions.



3170 Normally a job progresses from left to right. Other state transitions are unlikely, but are not forbidden.  
 3171 Not shown are the transitions to the 'canceled' state from the 'pending', 'pending-held', and 'processing-  
 3172 stopped' states.

3173 Jobs reach one of the three terminal states: 'completed', 'canceled', or 'aborted', after the jobs have  
 3174 completed all activity, including stacking output media, after the jobs have completed all activity, and all  
 3175 job status attributes have reached their final values for the job.

3176 Note: As with all other IPP attributes, if the implementation can not determine the correct value for this  
 3177 attribute, it SHOULD respond with the out-of-band value 'unknown' (see section 4.1) rather than try to  
 3178 guess at some possibly incorrect value and give the end user the wrong impression about the state of the  
 3179 Job object. For example, if the implementation is just a gateway into some printing system that does not  
 3180 provide detailed status about the print job, the IPP Job object's state might literally be 'unknown'.

## 3181 4.3.7.1 Partitioning of Job States

3182 This section partitions the 7 job states into phases: Job Not Completed, Job Retention, Job History, and  
3183 Job Removal. This section also explains the 'job-restartable' value of the "job-state-reasons" Job  
3184 Description attribute for use with the Restart-Job operation.

3185 Job Not Completed: When a job is in the 'pending', 'pending-held', 'processing', or 'processing-stopped'  
3186 states, the job is not completed.

3187 Job Retention: When a job enters one of the three terminal job states: 'completed', 'canceled', or  
3188 'aborted', the IPP Printer object MAY "retain" the job in a restartable condition for an implementation-  
3189 defined time period. This time period MAY be zero seconds and MAY depend on the terminal job state.  
3190 This phase is called Job Retention. While in the Job Retention phase, the job's document data is retained  
3191 and a client may restart the job using the Restart-Job operation. If the IPP object supports the "job-state-  
3192 reasons" attribute and the Restart-Job operation, then it SHOULD indicate that the job is restartable by  
3193 adding the 'job-restartable' value to the job's "job-state-reasons" attribute (see Section 4.3.8) during the  
3194 Job Retention phase.

3195 Job History: After the Job Retention phase expires for a job, the Printer object deletes the document  
3196 data for the job and the job becomes part of the Job History. The Printer object MAY also delete any  
3197 number of the job attributes. Since the job is no longer restartable, the Printer object MUST remove the  
3198 'job-restartable' value from the job's "job-state-reasons" attribute, if present.

3199 Job Removal: After the job has remained in the Job History for an implementation-defined time, such as  
3200 when the number of jobs exceeds a fixed number or after a fixed time period (which MAY be zero  
3201 seconds), the IPP Printer removes the job from the system.

3202 Using the Get-Jobs operation and supplying the 'not-completed' value for the "which-jobs" operation  
3203 attribute, a client is requesting jobs in the Job Not Completed phase. Using the Get-Jobs operation and  
3204 supplying the 'completed' value for the "which-jobs" operation attribute, a client is requesting jobs in the  
3205 Job Retention and Job History phases. Using the Get-Job-Attributes operation, a client is requesting a  
3206 job in any phase except Job Removal. After Job Removal, the Get-Job-Attributes and Get-Jobs  
3207 operations no longer are capable of returning any information about a job.

## 3208 4.3.8 job-state-reasons (1setOf type2 keyword)

3209 This attribute provides additional information about the job's current state, i.e., information that  
3210 augments the value of the job's "job-state" attribute.

3211 Implementation of these values is OPTIONAL, i.e., a Printer NEED NOT implement them, even if (1)  
3212 the output device supports the functionality represented by the reason and (2) is available to the Printer  
3213 object implementation. These values MAY be used with any job state or states for which the reason  
3214 makes sense. Furthermore, when implemented, the Printer MUST return these values when the reason  
3215 applies and MUST NOT return them when the reason no longer applies whether the value of the Job's  
3216 "job-state" attribute changed or not. When the Job does not have any reasons for being in its current  
3217 state, the value of the Job's "job-state-reasons" attribute MUST be 'none'.

3218 Note: While values cannot be added to the 'job-state' attribute without impacting deployed clients that  
3219 take actions upon receiving "job-state" values, it is the intent that additional "job-state-reasons" values  
3220 can be defined and registered without impacting such deployed clients. In other words, the "job-state-  
3221 reasons" attribute is intended to be extensible.

3222 The following standard keyword values are defined. For ease of understanding, the values are presented  
3223 in the order in which the reasons are likely to occur (if implemented), starting with the 'job-incoming'  
3224 value:

3225 'none': There are no reasons for the job's current state.

3226 'job-incoming': The Create-Job operation has been accepted by the Printer, but the Printer is  
3227 expecting additional Send-Document and/or Send-URI operations and/or is accessing/accepting  
3228 document data.

3229 'submission-interrupted': The job was not completely submitted for some unforeseen reason, such  
3230 as: (1) the Printer has crashed before the job was closed by the client, (2) the Printer or the  
3231 document transfer method has crashed in some non-recoverable way before the document data  
3232 was entirely transferred to the Printer, (3) the client crashed or failed to close the job before the  
3233 time-out period. See section 4.4.28.

3234 'job-outgoing': The Printer is transmitting the job to the output device.

3235 'job-hold-until-specified': The value of the job's "job-hold-until" attribute was specified with a time  
3236 period that is still in the future. The job MUST NOT be a candidate for processing until this  
3237 reason is removed and there are no other reasons to hold the job.

3238 'resources-are-not-ready': At least one of the resources needed by the job, such as media, fonts,  
3239 resource objects, etc., is not ready on any of the physical printer's for which the job is a candidate.  
3240 This condition MAY be detected when the job is accepted, or subsequently while the job is  
3241 pending or processing, depending on implementation. The job may remain in its current state or  
3242 be moved to the 'pending-held' state, depending on implementation and/or job scheduling policy.

3243 'printer-stopped-partly': The value of the Printer's "printer-state-reasons" attribute contains the value  
3244 'stopped-partly'.

3245 'printer-stopped': The value of the Printer's "printer-state" attribute is 'stopped'.

3246 'job-interpreting': Job is in the 'processing' state, but more specifically, the Printer is interpreting the  
3247 document data.

3248 'job-queued': Job is in the 'processing' state, but more specifically, the Printer has queued the  
3249 document data.

3250 'job-transforming': Job is in the 'processing' state, but more specifically, the Printer is interpreting  
3251 document data and producing another electronic representation.

3252 'job-printing': The output device is marking media. This value is useful for Printers which spend a  
3253 great deal of time processing (1) when no marking is happening and then want to show that  
3254 marking is now happening or (2) when the job is in the process of being canceled or aborted  
3255 while the job remains in the 'processing' state, but the marking has not yet stopped so that  
3256 impression or sheet counts are still increasing for the job.

3257 'job-canceled-by-user': The job was canceled by the owner of the job using the Cancel-Job request,  
3258 i.e., by a user whose authenticated identity is the same as the value of the originating user that  
3259 created the Job object, or by some other authorized end-user, such as a member of the job  
3260 owner's security group.

3261 'job-canceled-by-operator': The job was canceled by the operator using the Cancel-Job request, i.e.,  
3262 by a user who has been authenticated as having operator privileges (whether local or remote). If  
3263 the security policy is to allow anyone to cancel anyone's job, then this value may be used when  
3264 the job is canceled by other than the owner of the job. For such a security policy, in effect,  
3265 everyone is an operator as far as canceling jobs with IPP is concerned.  
3266 'job-canceled-at-device': The job was canceled by an unidentified local user, i.e., a user at a console  
3267 at the device.  
3268 'aborted-by-system': The job (1) is in the process of being aborted, (2) has been aborted by the  
3269 system and placed in the 'aborted' state, or (3) has been aborted by the system and placed in the  
3270 'pending-held' state, so that a user or operator can manually try the job again.  
3271 'processing-to-stop-point': The requester has issued a Cancel-job operation or the Printer object has  
3272 aborted the job, but is still performing some actions on the job until a specified stop point occurs  
3273 or job termination/cleanup is completed.  
3274

3275 This reason is recommended to be used in conjunction with the 'processing' job state to indicate  
3276 that the Printer object is still performing some actions on the job while the job remains in the  
3277 'processing' state. After all the job's job description attributes have stopped incrementing, the  
3278 Printer object moves the job from the 'processing' state to the 'canceled' or 'aborted' job states.

3279

3280 'service-off-line': The Printer is off-line and accepting no jobs. All 'pending' jobs are put into the  
3281 'pending-held' state. This situation could be true if the service's or document transform's input is  
3282 impaired or broken.

3283 'job-completed-successfully': The job completed successfully.

3284 'job-completed-with-warnings': The job completed with warnings.

3285 'job-completed-with-errors': The job completed with errors (and possibly warnings too).

3286 'job-restartable' - This job is retained (see section 4.3.7.1) and is currently able to be restarted using  
3287 the Restart-Job operation (see section 3.3.7). If 'job-restartable' is a value of the job's 'job-state-  
3288 reasons' attribute, then the IPP object MUST accept a Restart-Job operation for that job.

#### 3289 4.3.9 job-state-message (text(MAX))

3290 This attribute specifies information about the "job-state" and "job-state-reasons" attributes in human  
3291 readable text. If the Printer object supports this attribute, the Printer object MUST be able to generate  
3292 this message in any of the natural languages identified by the Printer's "generated-natural-language-  
3293 supported" attribute (see the "attributes-natural-language" operation attribute specified in Section  
3294 3.1.4.1).

3295 Note: the value SHOULD NOT contain additional information not contained in the values of the "job-  
3296 state" and "job-states-reasons" attributes, such as interpreter error information. Otherwise, application  
3297 programs might attempt to parse the (localized text). For such additional information such as interpreter  
3298 errors for application program consumption, a new attribute with keyword values, needs to be developed  
3299 and registered.

## 3300 4.3.10 number-of-documents (integer(0:MAX))

3301 This attribute indicates the number of documents in the job, i.e., the number of Send-Document, Send-  
3302 URI, Print-Job, or Print-URI operations that the Printer has accepted for this job, regardless of whether  
3303 the document data has reached the Printer object or not.

3304 Implementations supporting the OPTIONAL Create-Job/Send-Document/Send-URI operations  
3305 SHOULD support this attribute so that clients can query the number of documents in each job.

## 3306 4.3.11 output-device-assigned (name(127))

3307 This attribute identifies the output device to which the Printer object has assigned this job. If an output  
3308 device implements an embedded Printer object, the Printer object NEED NOT set this attribute. If a  
3309 print server implements a Printer object, the value MAY be empty (zero-length string) or not returned  
3310 until the Printer object assigns an output device to the job. This attribute is particularly useful when a  
3311 single Printer object support multiple devices (so called "fan-out").

## 3312 4.3.12 time-at-creation (integer(0:MAX))

3313 This attribute indicates the point in time at which the Job object was created. In order to populate this  
3314 attribute, the Printer object uses the value in its "printer-up-time" attribute at the time the Job object is  
3315 created.

## 3316 4.3.13 time-at-processing (integer(0:MAX))

3317 This attribute indicates the point in time at which the Job object began processing. In order to populate  
3318 this attribute, the Printer object uses the value in its "printer-up-time" attribute at the time the Job object  
3319 is moved into the 'processing' state for the first time.

## 3320 4.3.14 time-at-completed (integer(0:MAX))

3321 This attribute indicates the point in time at which the Job object completed (or was cancelled or aborted).  
3322 In order to populate this attribute, the Printer object uses the value in its "printer-up-time" attribute at the  
3323 time the Job object is moved into the 'completed' or 'canceled' or 'aborted' state.

## 3324 4.3.15 number-of-intervening-jobs (integer(0:MAX))

3325 This attribute indicates the number of jobs that are "ahead" of this job in the relative chronological order  
3326 of expected time to complete (i.e., the current scheduled order). For efficiency, it is only necessary to  
3327 calculate this value when an operation is performed that requests this attribute.

## 3328 4.3.16 job-message-from-operator (text(127))

3329 This attribute provides a message from an operator, system administrator or "intelligent" process to  
3330 indicate to the end user the reasons for modification or other management action taken on a job.

## 3331 4.3.17 job-k-octets (integer(0:MAX))

3332 This attribute specifies the total size of the document(s) in K octets, i.e., in units of 1024 octets requested  
3333 to be processed in the job. The value MUST be rounded up, so that a job between 1 and 1024 octets  
3334 MUST be indicated as being 1, 1025 to 2048 MUST be 2, etc.

3335 This value MUST NOT include the multiplicative factors contributed by the number of copies specified  
3336 by the "copies" attribute, independent of whether the device can process multiple copies without making  
3337 multiple passes over the job or document data and independent of whether the output is collated or not.  
3338 Thus the value is independent of the implementation and indicates the size of the document(s) measured  
3339 in K octets independent of the number of copies.

3340 This value MUST also not include the multiplicative factor due to a copies instruction embedded in the  
3341 document data. If the document data actually includes replications of the document data, this value will  
3342 include such replication. In other words, this value is always the size of the source document data, rather  
3343 than a measure of the hardcopy output to be produced.

3344 Note: This attribute and the following two attributes ("job-impressions" and "job-media-sheets") are not  
3345 intended to be counters; they are intended to be useful routing and scheduling information if known. For  
3346 these three attributes, the Printer object may try to compute the value if it is not supplied in the create  
3347 request. Even if the client does supply a value for these three attributes in the create request, the Printer  
3348 object MAY choose to change the value if the Printer object is able to compute a value which is more  
3349 accurate than the client supplied value. The Printer object may be able to determine the correct value for  
3350 these three attributes either right at job submission time or at any later point in time.

## 3351 4.3.18 job-impressions (integer(0:MAX))

3352 This attribute specifies the total size in number of impressions of the document(s) being submitted (see  
3353 the definition of impression in section 13.2.5).

3354 As with "job-k-octets", this value MUST NOT include the multiplicative factors contributed by the  
3355 number of copies specified by the "copies" attribute, independent of whether the device can process  
3356 multiple copies without making multiple passes over the job or document data and independent of  
3357 whether the output is collated or not. Thus the value is independent of the implementation and reflects  
3358 the size of the document(s) measured in impressions independent of the number of copies.

3359 As with "job-k-octets", this value MUST also not include the multiplicative factor due to a copies  
3360 instruction embedded in the document data. If the document data actually includes replications of the  
3361 document data, this value will include such replication. In other words, this value is always the number  
3362 of impressions in the source document data, rather than a measure of the number of impressions to be  
3363 produced by the job.

3364 See the Note in the "job-k-octets" attribute that also applies to this attribute.

## 3365 4.3.19 job-media-sheets (integer(0:MAX))

3366 This attribute specifies the total number of media sheets to be produced for this job.

3367 Unlike the "job-k-octets" and the "job-impressions" attributes, this value MUST include the  
3368 multiplicative factors contributed by the number of copies specified by the "copies" attribute and a  
3369 'number of copies' instruction embedded in the document data, if any. This difference allows the system  
3370 administrator to control the lower and upper bounds of both (1) the size of the document(s) with "job-k-  
3371 octets-supported" and "job-impressions-supported" and (2) the size of the job with "job-media-sheets-  
3372 supported".

3373 See the Note in the "job-k-octets" attribute that also applies to this attribute.

## 3374 4.3.20 job-k-octets-processed (integer(0:MAX))

3375 This attribute specifies the total number of octets processed in K octets, i.e., in units of 1024 octets so  
3376 far. The value MUST be rounded up, so that a job between 1 and 1024 octets inclusive MUST be  
3377 indicated as being 1, 1025 to 2048 inclusive MUST be 2, etc.

3378 For implementations where multiple copies are produced by the interpreter with only a single pass over  
3379 the data, the final value MUST be equal to the value of the "job-k-octets" attribute. For implementations  
3380 where multiple copies are produced by the interpreter by processing the data for each copy, the final  
3381 value MUST be a multiple of the value of the "job-k-octets" attribute.

3382 Note: This attribute and the following two attributes ("job-impressions-completed" and "job-sheets-  
3383 completed") are intended to be counters. That is, the value for a job that has not started processing  
3384 MUST be 0. When the job's "job-state" is 'processing' or 'processing-stopped', this value is intended to  
3385 contain the amount of the job that has been processed to the time at which the attributes are requested.

## 3386 4.3.21 job-impressions-completed (integer(0:MAX))

3387 This job attribute specifies the number of impressions completed for the job so far. For printing devices,  
3388 the impressions completed includes interpreting, marking, and stacking the output.

3389 See the note in "job-k-octets-processed" which also applies to this attribute.

## 3390 4.3.22 job-media-sheets-completed (integer(0:MAX))

3391 This job attribute specifies the media-sheets completed marking and stacking for the entire job so far  
3392 whether those sheets have been processed on one side or on both.

3393 See the note in "job-k-octets-processed" which also applies to this attribute.

## 3394 4.3.23 attributes-charset (charset)

3395 This REQUIRED attribute is populated using the value in the client supplied "attributes-charset"  
3396 attribute in the create request. It identifies the charset (coded character set and encoding method) used  
3397 by any Job attributes with attribute syntax 'text' and 'name' that were supplied by the client in the create  
3398 request. See Section 3.1.4 for a complete description of the "attributes-charset" operation attribute.

3399 This attribute does not indicate the charset in which the 'text' and 'name' values are stored internally in  
3400 the Job object. The internal charset is implementation-defined. The IPP object MUST convert from  
3401 whatever the internal charset is to that being requested in an operation as specified in Section 3.1.4.

## 3402 4.3.24 attributes-natural-language (naturalLanguage)

3403 This REQUIRED attribute is populated using the value in the client supplied "attributes-natural-  
3404 language" attribute in the create request. It identifies the natural language used for any Job attributes  
3405 with attribute syntax 'text' and 'name' that were supplied by the client in the create request. See Section  
3406 3.1.4 for a complete description of the "attributes-natural-language" operation attribute. See Sections  
3407 4.1.1.2 and 4.1.2.2 for how a Natural Language Override may be supplied explicitly for each 'text' and  
3408 'name' attribute value that differs from the value identified by the "attributes-natural-language" attribute.

## 3409 4.4 Printer Description Attributes

3410 These attributes form the attribute group called "printer-description". The following table summarizes  
3411 these attributes, their syntax, and whether or not they are REQUIRED for a Printer object to support. If  
3412 they are not indicated as REQUIRED, they are OPTIONAL. The maximum size in octets for 'text' and  
3413 'name' attributes is indicated in parentheses.

3414 Note: How these attributes are set by an Administrator is outside the scope of this IPP/1.1 document.



3465	color-supported	boolean		
3466	+-----+	+-----+	+-----+	+-----+
3467	reference-uri-schemes-	1setOf uriScheme		
3468	supported			
3469	+-----+	+-----+	+-----+	+-----+
3470	pdl-override-supported	type2 keyword		REQUIRED
3471	+-----+	+-----+	+-----+	+-----+
3472	printer-up-time	integer (1:MAX)		REQUIRED
3473	+-----+	+-----+	+-----+	+-----+
3474	printer-current-time	dateTime		
3475	+-----+	+-----+	+-----+	+-----+
3476	multiple-operation-time-out	integer (1:MAX)		
3477	+-----+	+-----+	+-----+	+-----+
3478	compression-supported	1setOf type3 keyword		
3479	+-----+	+-----+	+-----+	+-----+
3480	job-k-octets-supported	rangeOfInteger (0:MAX)		
3481	+-----+	+-----+	+-----+	+-----+
3482	job-impressions-supported	rangeOfInteger (0:MAX)		
3483	+-----+	+-----+	+-----+	+-----+
3484	job-media-sheets-supported	rangeOfInteger (0:MAX)		
3485	+-----+	+-----+	+-----+	+-----+
3486	pages-per-minute	integer(0:MAX)		
3487	+-----+	+-----+	+-----+	+-----+
3488	pages-per-minute-color	integer(0:MAX)		
3489	+-----+	+-----+	+-----+	+-----+
3490				

#### 3491 4.4.1 printer-uri-supported (1setOf uri)

3492 This REQUIRED Printer attribute contains at least one URI for the Printer object. It OPTIONALLY  
 3493 contains more than one URI for the Printer object. An administrator determines a Printer object's  
 3494 URI(s) and configures this attribute to contain those URIs by some means outside the scope of this  
 3495 IPP/1.1 document. The precise format of this URI is implementation dependent and depends on the  
 3496 protocol. See the next section for a description "uri-security-supported" which is the REQUIRED  
 3497 companion attribute to this "printer-uri-supported" attribute. See section 2.4 on Printer object identity  
 3498 and section 8.2 on security and URIs for more information.

#### 3499 4.4.2 uri-security-supported (1setOf type2 keyword)

3500 This REQUIRED Printer attribute MUST have the same cardinality (contain the same number of values)  
 3501 as the "printer-uri-supported" attribute. This attribute identifies the security mechanisms used for each  
 3502 URI listed in the "printer-uri-supported" attribute. The "i th" value in "uri-security-supported"  
 3503 corresponds to the "i th" value in "printer-uri-supported" and it describes the security mechanisms used  
 3504 for accessing the Printer object via that URI. The following standard values are defined:

- 3505 'none': There are no secure communication channel protocols in use for the given URI.
- 3506 'ssl3': SSL3 [SSL] is the secure communications channel protocol in use for the given URI. For use  
 3507 in IPP/1.0.

3508       `tls`: TLS [RFC2246] is the secure communications channel protocol in use for the given URI. For  
3509       use in IPP/1.1.

3510

3511 Consider the following example. For a single Printer object, an administrator configures the "printer-uri-  
3512 supported" and "uri-security-supported" attributes as follows:

3513       "printer-uri-supported": `xxx://acme.com/open-use-printer`, `xxx://acme.com/restricted-use-printer`,  
3514       `xxx://acme.com/private-printer`  
3515       "uri-security-supported": `none`, `none`, `tls`

3516

3517 Note: `xxx` is not a valid scheme. See the IPP/1.1 "Transport and Encoding" specification [ipp-pro] for  
3518 the actual URI schemes to be used in object target attributes.

3519 In this case, one Printer object has three URIs.

- 3520       - For the first URI, `xxx://acme.com/open-use-printer`, the value `none` in "uri-security-supported"  
3521       indicates that there is no secure channel protocol configured to run under HTTP. The name  
3522       implies that there is no Basic or Digest authentication being used, but it is up to the client to  
3523       determine that while using HTTP underneath the IPP application protocol.
  - 3524       - For the second URI, `xxx://acme.com/restricted-use-printer`, the value `none` in "uri-security-  
3525       supported" indicates that there is no secure channel protocol configured to run under HTTP. In  
3526       this case, although the name does imply that there is some sort of Basic or Digest authentication  
3527       being used within HTTP, it is up to the client to determine that while using HTTP and by  
3528       processing any `401 Unauthorized` HTTP error messages.
  - 3529       - For the third URI, `xxx://acme.com/private-printer`, the value `tls` in "uri-security-supported"  
3530       indicates that TLS is being used to secure the channel. The client SHOULD be prepared to use  
3531       TLS framing to negotiate an acceptable ciphersuite to use while communicating with the Printer  
3532       object. In this case, the name implies the use of a secure communications channel, but the fact is  
3533       made explicit by the presence of the `tls` value in "uri-security-supported". The client does not  
3534       need to resort to understanding which security it must use by following naming conventions or by  
3535       parsing the URI to determine which security mechanisms are implied.
- 3536

3537 It is expected that many IPP Printer objects will be configured to support only one channel (either  
3538 configured to use TLS access or not), and will therefore only ever have one URI listed in the "printer-uri-  
3539 supported" attribute. No matter the configuration of the Printer object (whether it has only one URI or  
3540 more than one URI), a client MUST supply only one URI in the target "printer-uri" operation attribute.

#### 3541 4.4.3 printer-name (name(127))

3542 This REQUIRED Printer attribute contains the name of the Printer object. It is a name that is more end-  
3543 user friendly than a URI. An administrator determines a printer's name and sets this attribute to that  
3544 name. This name may be the last part of the printer's URI or it may be unrelated. In non-US-English  
3545 locales, a name may contain characters that are not allowed in a URI.

## 3546 4.4.4 printer-location (text(127))

3547 This Printer attribute identifies the location of the device. This could include things like: "in Room  
3548 123A, second floor of building XYZ".

## 3549 4.4.5 printer-info (text(127))

3550 This Printer attribute identifies the descriptive information about this Printer object. This could include  
3551 things like: "This printer can be used for printing color transparencies for HR presentations", or "Out of  
3552 courtesy for others, please print only small (1-5 page) jobs at this printer", or even "This printer is going  
3553 away on July 1, 1997, please find a new printer".

## 3554 4.4.6 printer-more-info (uri)

3555 This Printer attribute contains a URI used to obtain more information about this specific Printer object.  
3556 For example, this could be an HTTP type URI referencing an HTML page accessible to a Web Browser.  
3557 The information obtained from this URI is intended for end user consumption. Features outside the  
3558 scope of IPP can be accessed from this URI. The information is intended to be specific to this printer  
3559 instance and site specific services (e.g. job pricing, services offered, end user assistance). The device  
3560 manufacturer may initially populate this attribute.

## 3561 4.4.7 printer-driver-installer (uri)

3562 This Printer attribute contains a URI to use to locate the driver installer for this Printer object. This  
3563 attribute is intended for consumption by automata. The mechanics of print driver installation is outside  
3564 the scope of this IPP/1.1 document. The device manufacturer may initially populate this attribute.

## 3565 4.4.8 printer-make-and-model (text(127))

3566 This Printer attribute identifies the make and model of the device. The device manufacturer may  
3567 initially populate this attribute.

## 3568 4.4.9 printer-more-info-manufacturer (uri)

3569 This Printer attribute contains a URI used to obtain more information about this type of device. The  
3570 information obtained from this URI is intended for end user consumption. Features outside the scope of  
3571 IPP can be accessed from this URI (e.g., latest firmware, upgrades, print drivers, optional features  
3572 available, details on color support). The information is intended to be germane to this printer without  
3573 regard to site specific modifications or services. The device manufacturer may initially populate this  
3574 attribute.

## 3575 4.4.10 printer-state (type1 enum)

3576 This REQUIRED Printer attribute identifies the current state of the device. The "printer-state reasons"  
 3577 attribute augments the "printer-state" attribute to give more detailed information about the Printer in the  
 3578 given printer state.

3579 A Printer object need only update this attribute before responding to an operation which requests the  
 3580 attribute; the Printer object NEED NOT update this attribute continually, since asynchronous event  
 3581 notification is not part of IPP/1.1. A Printer NEED NOT implement all values if they are not applicable  
 3582 to a given implementation.

3583 The following standard enum values are defined:

3584 Value	Symbolic Name and Description
3585 3586 '3'	'idle': If a Printer receives a job (whose required resources are ready) while in this state, such a job MUST transit into the 'processing' state immediately. If the "printer-state-reasons" attribute contains any reasons, they MUST be reasons that would not prevent a job from transiting into the 'processing' state immediately, e.g., 'toner-low'. Note: if a Printer controls more than one output device, the above definition implies that a Printer is 'idle' if at least one output device is idle.
3592 3593 '4'	'processing': If a Printer receives a job (whose required resources are ready) while in this state, such a job MUST transit into the 'pending' state immediately. Such a job MUST transit into the 'processing' state only after jobs ahead of it complete. If the "printer-state-reasons" attribute contains any reasons, they MUST be reasons that do not prevent the current job from printing, e.g. 'toner-low'. Note: if a Printer controls more than one output device, the above definition implies that a Printer is 'processing' if at least one output device is processing, and none is idle.
3600 3601 '5'	'stopped': If a Printer receives a job (whose required resources are ready) while in this state, such a job MUST transit into the 'pending' state immediately. Such a job MUST transit into the 'processing' state only after some human fixes the problem that stopped the printer and after jobs ahead of it complete processing. If supported, the "printer-state-reasons" attribute MUST contain at least one reason, e.g. 'media-jam', which prevents it from either processing the current job or transitioning a 'pending' job to the 'processing' state.
3608 3609	Note: if a Printer controls more than one output device, the above definition implies that a Printer is 'stopped' only if all output devices are stopped. Also, it is tempting to define 'stopped' as when a sufficient number of output devices are stopped and leave it to an implementation to define the sufficient number. But such a rule complicates the definition of 'stopped' and 'processing'. For example, with this alternate definition of 'stopped', a job can move from 'pending' to 'processing' without human intervention, even though the Printer is stopped.
3610 3611 3612 3613 3614 3615 3616	

## 3617 4.4.11 printer-state-reasons (1setOf type2 keyword)

3618 This Printer attribute supplies additional detail about the device's state.

3619 Each keyword value MAY have a suffix to indicate its level of severity. The three levels are: report  
3620 (least severe), warning, and error (most severe).

3621 - 'report': This suffix indicates that the reason is a "report". An implementation may choose to omit  
3622 some or all reports. Some reports specify finer granularity about the printer state; others serve as  
3623 a precursor to a warning. A report MUST contain nothing that could affect the printed output.

3624 - 'warning': This suffix indicates that the reason is a "warning". An implementation may choose to  
3625 omit some or all warnings. Warnings serve as a precursor to an error. A warning MUST contain  
3626 nothing that prevents a job from completing, though in some cases the output may be of lower  
3627 quality.

3628 - 'error': This suffix indicates that the reason is an "error". An implementation MUST include all  
3629 errors. If this attribute contains one or more errors, printer MUST be in the stopped state.  
3630

3631 If the implementation does not add any one of the three suffixes, all parties MUST assume that the  
3632 reason is an "error".

3633 If a Printer object controls more than one output device, each value of this attribute MAY apply to one or  
3634 more of the output devices. An error on one output device that does not stop the Printer object as a  
3635 whole MAY appear as a warning in the Printer's "printer-state-reasons attribute". If the "printer-state"  
3636 for such a Printer has a value of 'stopped', then there MUST be an error reason among the values in the  
3637 "printer-state-reasons" attribute.

3638 The following standard keyword values are defined:

3639 'other': The device has detected an error other than one listed in this document.

3640 'none': There are not reasons. This state reason is semantically equivalent to "printer-state-reasons"  
3641 without any value.

3642 'media-needed': A tray has run out of media.

3643 'media-jam': The device has a media jam.

3644 'moving-to-paused': Someone has paused the Printer object using the Pause-Printer operation (see  
3645 section 3.2.7) or other means, but the device(s) are taking an appreciable time to stop. Later,  
3646 when all output has stopped, the "printer-state" becomes 'stopped', and the 'paused' value replaces  
3647 the 'moving-to-paused' value in the "printer-state-reasons" attribute.

3648 'paused': Someone has paused the Printer object using the Pause-Printer operation (see section 3.2.7)  
3649 or other means and the Printer object's "printer-state" is 'stopped'. In this state, a Printer MUST  
3650 NOT produce printed output, but it MUST perform other operations requested by a client. If a  
3651 Printer had been printing a job when the Printer was paused, the Printer MUST resume printing  
3652 that job when the Printer is no longer paused and leave no evidence in the printed output of such  
3653 a pause.

3654 'shutdown': Someone has removed a Printer object from service, and the device may be powered  
3655 down or physically removed. In this state, a Printer object MUST NOT produce printed output,  
3656 and unless the Printer object is realized by a print server that is still active, the Printer object  
3657 MUST perform no other operations requested by a client, including returning this value. If a

3658 Printer object had been printing a job when it was shutdown, the Printer NEED NOT resume  
3659 printing that job when the Printer is no longer shutdown. If the Printer resumes printing such a  
3660 job, it may leave evidence in the printed output of such a shutdown, e.g. the part printed before  
3661 the shutdown may be printed a second time after the shutdown.

3662 'connecting-to-device': The Printer object has scheduled a job on the output device and is in the  
3663 process of connecting to a shared network output device (and might not be able to actually start  
3664 printing the job for an arbitrarily long time depending on the usage of the output device by other  
3665 servers on the network).

3666 'timed-out': The server was able to connect to the output device (or is always connected), but was  
3667 unable to get a response from the output device.

3668 'stopping': The Printer object is in the process of stopping the device and will be stopped in a while.  
3669 When the device is stopped, the Printer object will change the Printer object's state to 'stopped'.  
3670 The 'stopping-warning' reason is never an error, even for a Printer with a single output device.  
3671 When an output-device ceases accepting jobs, the Printer will have this reason while the output  
3672 device completes printing.

3673 'stopped-partly': When a Printer object controls more than one output device, this reason indicates  
3674 that one or more output devices are stopped. If the reason is a report, fewer than half of the  
3675 output devices are stopped. If the reason is a warning, fewer than all of the output devices are  
3676 stopped.

3677 'toner-low': The device is low on toner.

3678 'toner-empty': The device is out of toner.

3679 'spool-area-full': The limit of persistent storage allocated for spooling has been reached.

3680 'cover-open': One or more covers on the device are open.

3681 'interlock-open': One or more interlock devices on the printer are unlocked.

3682 'door-open': One or more doors on the device are open.

3683 'input-tray-missing': One or more input trays are not in the device.

3684 'media-low': At least one input tray is low on media.

3685 'media-empty': At least one input tray is empty.

3686 'output-tray-missing': One or more output trays are not in the device

3687 'output-area-almost-full': One or more output area is almost full (e.g. tray, stacker, collator).

3688 'output-area-full': One or more output area is full. (e.g. tray, stacker, collator)

3689 'marker-supply-low': The device is low on at least one marker supply. (e.g. toner, ink, ribbon)

3690 'marker-supply-empty': The device is out of at least one marker supply. (e.g. toner, ink, ribbon)

3691 'marker-waste-almost-full': The device marker supply waste receptacle is almost full.

3692 'marker-waste-full': The device marker supply waste receptacle is full.

3693 'fuser-over-temp': The fuser temperature is above normal.

3694 'fuser-under-temp': The fuser temperature is below normal.

3695 'opc-near-eol': The optical photo conductor is near end of life.

3696 'opc-life-over': The optical photo conductor is no longer functioning.

3697 'developer-low': The device is low on developer.

3698 'developer-empty': The device is out of developer.

3699 'interpreter-resource-unavailable': An interpreter resource is unavailable (i.e. font, form)

3700

## 3701 4.4.12 printer-state-message (text(MAX))

3702 This Printer attribute specifies the additional information about the printer state and printer state reasons  
 3703 in human readable text. If the Printer object supports this attribute, the Printer object MUST be able to  
 3704 generate this message in any of the natural languages identified by the Printer's "generated-natural-  
 3705 language-supported" attribute (see the "attributes-natural-language" operation attribute specified in  
 3706 Section 3.1.4.1).

## 3707 4.4.13 operations-supported (1setOf type2 enum)

3708 This REQUIRED Printer attribute specifies the set of supported operations for this Printer object and  
 3709 contained Job objects.

3710 Note: This attribute is encoded as any other enum attribute syntax according to [IPP-PRO] as 32-bits.  
 3711 However, all 32-bit enum values for this attribute MUST NOT exceed 0x00008FFF, since these same  
 3712 values are also passed in two octets in the "operation-id" parameter (see section 3.1.1) in each Protocol  
 3713 request with the two high order octets omitted in order to indicate the operation being performed [IPP-  
 3714 PRO].

3715 The following standard enum and "operation-id" (see section 3.1.2) values are defined:

3716	Value	Operation Name
3717	-----	-----
3718		
3719	0x0000	reserved, not used
3720	0x0001	reserved, not used
3721	0x0002	Print-Job
3722	0x0003	Print-URI
3723	0x0004	Validate-Job
3724	0x0005	Create-Job
3725	0x0006	Send-Document
3726	0x0007	Send-URI
3727	0x0008	Cancel-Job
3728	0x0009	Get-Job-Attributes
3729	0x000A	Get-Jobs
3730	0x000B	Get-Printer-Attributes
3731	0x000C	Hold-Job
3732	0x000D	Release-Job
3733	0x000E	Restart-Job
3734	0x000F	reserved for a future operation
3735	0x0010	Pause-Printer
3736	0x0011	Resume-Printer
3737	0x0012	Purge-Jobs
3738	0x00013-0x3FFF	reserved for future operations
3739	0x4000-0x8FFF	reserved for private extensions
3740		

3741 This allows for certain vendors to implement private extensions that are guaranteed to not conflict with  
3742 future registered extensions. However, there is no guarantee that two or more private extensions will not  
3743 conflict.

#### 3744 4.4.14 charset-configured (charset)

3745 This REQUIRED Printer attribute identifies the charset that the Printer object has been configured to  
3746 represent 'text' and 'name' Printer attributes that are set by the operator, system administrator, or  
3747 manufacturer, i.e., for "printer-name" (name), "printer-location" (text), "printer-info" (text), and "printer-  
3748 make-and-model" (text). Therefore, the value of the Printer object's "charset-configured" attribute  
3749 MUST also be among the values of the Printer object's "charset-supported" attribute.

#### 3750 4.4.15 charset-supported (1setOf charset)

3751 This REQUIRED Printer attribute identifies the set of charsets that the Printer and contained Job objects  
3752 support in attributes with attribute syntax 'text' and 'name'. At least the value 'utf-8' MUST be present,  
3753 since IPP objects MUST support the UTF-8 [RFC2279] charset. If a Printer object supports a charset, it  
3754 means that for all attributes of syntax 'text' and 'name' the IPP object MUST (1) accept the charset in  
3755 requests and return the charset in responses as needed.

3756 If more charsets than UTF-8 are supported, the IPP object MUST perform charset conversion between  
3757 the charsets as described in Section 3.2.1.2.

#### 3758 4.4.16 natural-language-configured (naturalLanguage)

3759 This REQUIRED Printer attribute identifies the natural language that the Printer object has been  
3760 configured to represent 'text' and 'name' Printer attributes that are set by the operator, system  
3761 administrator, or manufacturer, i.e., for "printer-name" (name), "printer-location" (text), "printer-info"  
3762 (text), and "printer-make-and-model" (text). When returning these Printer attributes, the Printer object  
3763 MAY return them in the configured natural language specified by this attribute, instead of the natural  
3764 language requested by the client in the "attributes-natural-language" operation attribute. See Section  
3765 3.1.4.1 for the specification of the OPTIONAL multiple natural language support. Therefore, the value  
3766 of the Printer object's "natural-language-configured" attribute MUST also be among the values of the  
3767 Printer object's "generated-natural-language-supported" attribute.

#### 3768 4.4.17 generated-natural-language-supported (1setOf naturalLanguage)

3769 This REQUIRED Printer attribute identifies the natural language(s) that the Printer object and contained  
3770 Job objects support in attributes with attribute syntax 'text' and 'name'. The natural language(s)  
3771 supported depends on implementation and/or configuration. Unlike charsets, IPP objects MUST accept  
3772 requests with any natural language or any Natural Language Override whether the natural language is  
3773 supported or not.

3774 If a Printer object supports a natural language, it means that for any of the attributes for which the Printer  
3775 or Job object generates messages, i.e., for the "job-state-message" and "printer-state-message" attributes

3776 and Operation Messages (see Section 3.1.5) in operation responses, the Printer and Job objects MUST be  
3777 able to generate messages in any of the Printer's supported natural languages. See section 3.1.4 for the  
3778 specification of 'text' and 'name' attributes in operation requests and responses.

3779 Note: A Printer object that supports multiple natural languages, often has separate catalogs of messages,  
3780 one for each natural language supported.

#### 3781 4.4.18 document-format-default (mimeMediaType)

3782 This REQUIRED Printer attribute identifies the document format that the Printer object has been  
3783 configured to assume if the client does not supply a "document-format" operation attribute in any of the  
3784 operation requests that supply document data. The standard values for this attribute are Internet Media  
3785 types (sometimes called MIME types). For further details see the description of the 'mimeMediaType'  
3786 attribute syntax in Section 4.1.9.

#### 3787 4.4.19 document-format-supported (1setOf mimeMediaType)

3788 This REQUIRED Printer attribute identifies the set of document formats that the Printer object and  
3789 contained Job objects can support. For further details see the description of the 'mimeMediaType'  
3790 attribute syntax in Section 4.1.9.

#### 3791 4.4.20 printer-is-accepting-jobs (boolean)

3792 This REQUIRED Printer attribute indicates whether the printer is currently able to accept jobs, i.e., is  
3793 accepting Print-Job, Print-URI, and Create-Job requests. If the value is 'true', the printer is accepting  
3794 jobs. If the value is 'false', the Printer object is currently rejecting any jobs submitted to it. In this case,  
3795 the Printer object returns the 'server-error-not-accepting-jobs' status code.

3796 Note: This value is independent of the "printer-state" and "printer-state-reasons" attributes because its  
3797 value does not affect the current job; rather it affects future jobs. This attribute may cause the Printer to  
3798 reject jobs when the "printer-state" is 'idle' or it may cause the Printer object to accept jobs when the  
3799 "printer-state" is 'stopped'.

#### 3800 4.4.21 queued-job-count (integer(0:MAX))

3801 This RECOMMENDED Printer attribute contains a count of the number of jobs that are either 'pending',  
3802 'processing', 'pending-held', or 'processing-stopped' and is set by the Printer object.

#### 3803 4.4.22 printer-message-from-operator (text(127))

3804 This Printer attribute provides a message from an operator, system administrator or "intelligent" process  
3805 to indicate to the end user information or status of the printer, such as why it is unavailable or when it is  
3806 expected to be available.

## 3807 4.4.23 color-supported (boolean)

3808 This Printer attribute identifies whether the device is capable of any type of color printing at all,  
3809 including highlight color. All document instructions having to do with color are embedded within the  
3810 document PDL (none are external IPP attributes in IPP/1.1).

3811 Note: end-users are able to determine the nature and details of the color support by querying the  
3812 "printer-more-info-manufacturer" Printer attribute.

## 3813 4.4.24 reference-uri-schemes-supported (1setOf uriScheme)

3814 This Printer attribute specifies which URI schemes are supported for use in the "document-uri" operation  
3815 attribute of the Print-URI or Send-URI operation. If a Printer object supports these optional operations,  
3816 it **MUST** support the "reference-uri-schemes-supported" Printer attribute with at least the following  
3817 schemed URI value:

3818     'ftp': The Printer object will use an FTP 'get' operation as defined in RFC 2228 [RFC2228] using  
3819     FTP URLs as defined by [RFC2396] and[RFC2316].  
3820

3821 The Printer object **MAY OPTIONALLY** support other URI schemes (see section 4.1.6).

## 3822 4.4.25 pdl-override-supported (type2 keyword)

3823 This **REQUIRED** Printer attribute expresses the ability for a particular Printer implementation to either  
3824 attempt to override document data instructions with IPP attributes or not.

3825 This attribute takes on the following values:

- 3826     - 'attempted': This value indicates that the Printer object attempts to make the IPP attribute values  
3827     take precedence over embedded instructions in the document data, however there is no guarantee.
- 3828     - 'not-attempted': This value indicates that the Printer object makes no attempt to make the IPP  
3829     attribute values take precedence over embedded instructions in the document data.  
3830

3831 Section 16 contains a full description of how this attribute interacts with and affects other IPP attributes,  
3832 especially the "ipp-attribute-fidelity" attribute.

## 3833 4.4.26 printer-up-time (integer(1:MAX))

3834 This **REQUIRED** Printer attribute indicates the amount of time (in seconds) that this instance of this  
3835 Printer implementation has been up and running. This value is used to populate the Job attributes "time-  
3836 at-creation", "time-at-processing", and "time-at-completed". These time values are all measured in  
3837 seconds and all have meaning only relative to this attribute, "printer-up-time". The value is a  
3838 monotonically increasing value starting from 1 when the Printer object is started-up (initialized, booted,  
3839 etc.).

3840 If the Printer object goes down at some value 'n', and comes back up, the implementation **MAY**:

- 3841 1. Know how long it has been down, and resume at some value greater than 'n', or  
3842 2. Restart from 1.  
3843

3844 In the first case, the Printer SHOULD not tweak any existing related Job attributes ("time-at-creation",  
3845 "time-at-processing", and "time-at-completed"). In the second case, the Printer object SHOULD reset  
3846 those attributes to 0. If a client queries a time-related Job attribute and finds the value to be 0, the client  
3847 MUST assume that the Job was submitted in some life other than the Printer's current life.

#### 3848 4.4.27 printer-current-time (dateTime)

3849 This Printer attribute indicates the current absolute wall-clock time. If an implementation supports this  
3850 attribute, then a client could calculate the absolute wall-clock time each Job's "time-at-creation", "time-  
3851 at-processing", and "time-at-completed" attributes by using both "printer-up-time" and this attribute,  
3852 "printer-current-time". If an implementation does not support this attribute, a client can only calculate  
3853 the relative time of certain events based on the REQUIRED "printer-up-time" attribute.

#### 3854 4.4.28 multiple-operation-time-out (integer(1:MAX))

3855 This Printer attributes identifies the minimum time (in seconds) that the Printer object waits for  
3856 additional Send-Document or Send-URI operations to follow a still-open multi-document Job object  
3857 before taking any recovery actions, such as the ones indicated in section 3.3.1. If the Printer object  
3858 supports the Create-Job operation (see section 3.2.4), it MUST support this attribute.

3859 It is RECOMMENDED that vendors supply a value for this attribute that is between 60 and 240  
3860 seconds. An implementation MAY allow a system administrator to set this attribute (by means outside  
3861 this IPP/1.1 document). If so, the system administrator MAY be able to set values outside this range.

#### 3862 4.4.29 compression-supported (1setOf type3 keyword)

3863 This Printer attribute identifies the set of supported compression algorithms for document data.  
3864 Compression only applies to the document data; compression does not apply to the encoding of the IPP  
3865 operation itself. The supported values are used to validate the client supplied "compression" operation  
3866 attributes in Print-Job, Send-Document, and Send-URI requests.

3867 Standard values are :

- 3868 'none': no compression is used.  
3869 'deflate': ZIP public domain inflate/deflate) compression technology  
3870 'gzip' GNU zip compression technology described in RFC 1952 [RFC1952].  
3871 'compress': UNIX compression technology  
3872

#### 3873 4.4.30 job-k-octets-supported (rangeOfInteger(0:MAX))

3874 This Printer attribute specifies the upper and lower bounds of total sizes of jobs in K octets, i.e., in units  
3875 of 1024 octets. The supported values are used to validate the client supplied "job-k-octets" operation

3876 attributes in create requests. The corresponding job description attribute "job-k-octets" is defined in  
3877 section 4.3.17.

#### 3878 4.4.31 job-impressions-supported (rangeOfInteger(0:MAX))

3879 This Printer attribute specifies the upper and lower bounds for the number of impressions per job. The  
3880 supported values are used to validate the client supplied "job-impressions" operation attributes in create  
3881 requests. The corresponding job description attribute "job-impressions" is defined in section 4.3.18.

#### 3882 4.4.32 job-media-sheets-supported (rangeOfInteger(0:MAX))

3883 This Printer attribute specifies the upper and lower bounds for the number of media sheets per job. The  
3884 supported values are used to validate the client supplied "job-media-sheets" operation attributes in create  
3885 requests. The corresponding Job attribute "job-media-sheets" is defined in section 4.3.19.

#### 3886 4.4.33 pages-per-minute (integer(0:MAX))

3887 This Printer attributes specifies the nominal number of pages per minute to the nearest whole number  
3888 which may be generated by this printer (e.g., simplex, black-and-white). This attribute is informative,  
3889 not a service guarantee. Generally, it is the value used in the marketing literature to describe the device.

3890 A value of 0 indicates a device that takes more than two minutes to process a page.

#### 3891 4.4.34 pages-per-minute-color (integer(0:MAX))

3892 This Printer attributes specifies the nominal number of pages per minute to the nearest whole number  
3893 which may be generated by this printer when printing color (e.g., simplex, color). For purposes of this  
3894 attribute, "color" means the same as for the "color-supported" attribute, namely, the device is capable of  
3895 any type of color printing at all, including highlight color. This attribute is informative, not a service  
3896 guarantee. Generally, it is the value used in the marketing literature to describe the color capabilities of  
3897 this device.

3898 A value of 0 indicates a device that takes more than two minutes to process a page.

3899 Note: If a color device has several color modes, it MAY use the pages-per-minute value for this  
3900 attribute that corresponds to the mode that produces the highest number.

3901 Black and white only printers MUST NOT support this attribute. If this attribute is present, then the  
3902 "color-supported" Printer description attribute MUST be present and have a 'true' value.

3903 Note: The values of these two attributes returned by the Get-Printer-Attributes operation MAY be  
3904 affected by the "document-format" attribute supplied by the client in the Get-Printer-Attributes request.  
3905 In other words, the implementation MAY have different speeds depending on the document format  
3906 being processed. See section 3.2.5.1 Get-Printer-Attributes.

## 3907 5. Conformance

3908 This section describes conformance issues and requirements. This document introduces model entities  
3909 such as objects, operations, attributes, attribute syntaxes, and attribute values. These conformance  
3910 sections describe the conformance requirements which apply to these model entities.

## 3911 5.1 Client Conformance Requirements

3912 A conforming client **MUST** support all **REQUIRED** operations as defined in this document. For each  
3913 attribute included in an operation request, a conforming client **MUST** supply a value whose type and  
3914 value syntax conforms to the requirements of the Model document as specified in Sections 3 and 3.3.5.  
3915 A conforming client **MAY** supply any registered extensions and/or private extensions in an operation  
3916 request, as long as they meet the requirements in Section 6.

3917 Otherwise, there are no conformance requirements placed on the user interfaces provided by IPP clients  
3918 or their applications. For example, one application might not allow an end user to submit multiple  
3919 documents per job, while another does. One application might first query a Printer object in order to  
3920 supply a graphical user interface (GUI) dialogue box with supported and default values whereas a  
3921 different implementation might not.

3922 When sending a request, an IPP client **NEED NOT** supply any attributes that are indicated as  
3923 **OPTIONALLY** supplied by the client.

3924 A client **MUST** be able to accept any of the attribute syntaxes defined in Section 4.1, including their full  
3925 range, that may be returned to it in a response from a Printer object. In particular for each attribute that  
3926 the client supports whose attribute syntax is 'text', the client **MUST** accept and process both the  
3927 'textWithoutLanguage' and 'textWithLanguage' forms. Similarly, for each attribute that the client  
3928 supports whose attribute syntax is 'name', the client **MUST** accept and process both the  
3929 'nameWithoutLanguage' and 'nameWithLanguage' forms. For presentation purposes, truncation of long  
3930 attribute values is not recommended. A recommended approach would be for the client implementation  
3931 to allow the user to scroll through long attribute values.

3932 A query response may contain attribute groups, attributes, and values that the client does not expect.  
3933 Therefore, a client implementation **MUST** gracefully handle such responses and not refuse to inter-  
3934 operate with a conforming Printer that is returning extended registered or private attributes and/or  
3935 attribute values that conform to Section 6. Clients may choose to ignore any parameters, attributes, or  
3936 values that they do not understand.

## 3937 5.2 IPP Object Conformance Requirements

3938 This section specifies the conformance requirements for conforming implementations with respect to  
3939 objects, operations, and attributes.

## 3940 5.2.1 Objects

3941 Conforming implementations **MUST** implement all of the model objects as defined in this specification  
3942 in the indicated sections:

3943 Section 2.1 - Printer Object

3944 Section 2.2 - Job Object

3945

## 3946 5.2.2 Operations

3947 Conforming IPP object implementations **MUST** implement all of the **REQUIRED** model operations,  
3948 including **REQUIRED** responses, as defined in this specification in the indicated sections:

3949 For a Printer object:

3950 Print-Job (section 3.2.1) **REQUIRED**

3951 Print-URI (section 3.2.2) **OPTIONAL**

3952 Validate-Job (section 3.2.3) **REQUIRED**

3953 Create-Job (section 3.2.4) **OPTIONAL**

3954 Get-Printer-Attributes (section 3.2.5) **REQUIRED**

3955 Get-Jobs (section 3.2.6) **REQUIRED**

3956 Pause-Printer (section 3.2.7) **OPTIONAL**

3957 Resume-Printer (section 3.2.8) **OPTIONAL**

3958 Purge-Jobs (section 3.2.9) **OPTIONAL**

3959

3960 For a Job object:

3961 Send-Document (section 3.3.1) **OPTIONAL**

3962 Send-URI (section 3.3.2) **OPTIONAL**

3963 Cancel-Job (section 3.3.3) **REQUIRED**

3964 Get-Job-Attributes (section 3.3.4) **REQUIRED**

3965 Hold-Job (section 3.3.5) **OPTIONAL**

3966 Release-Job (section 3.3.6) **OPTIONAL**

3967 Restart-Job (section 3.3.7) **OPTIONAL**

3968

3969 Conforming IPP objects **MUST** support all **REQUIRED** operation attributes and all values of such  
3970 attributes if so indicated in the description. Conforming IPP objects **MUST** ignore all unsupported or  
3971 unknown operation attributes or operation attribute groups received in a request, but **MUST** reject a  
3972 request that contains a supported operation attribute that contains an unsupported value.

3973 The following section on object attributes specifies the support required for object attributes.

## 3974 5.2.3 IPP Object Attributes

3975 Conforming IPP objects **MUST** support all of the **REQUIRED** object attributes, as defined in this  
3976 specification in the indicated sections.

3977 If an object supports an attribute, it **MUST** support only those values specified in this document or  
3978 through the extension mechanism described in section 5.2.4. It **MAY** support any non-empty subset of  
3979 these values. That is, it **MUST** support at least one of the specified values and at most all of them.

#### 3980 5.2.4 Versions

3981 Clients **MUST** support version 1.1 and **MAY** also support version 1.0. IPP objects **MUST** support both  
3982 version 1.0 and 1.1. See section 3.1.7.

#### 3983 5.2.5 Extensions

3984 A conforming IPP object **MAY** support registered extensions and private extensions, as long as they  
3985 meet the requirements specified in Section 6.

3986 For each attribute included in an operation response, a conforming IPP object **MUST** return a value  
3987 whose type and value syntax conforms to the requirement of the Model document as specified in  
3988 Sections 3 and 3.3.5.

#### 3989 5.2.6 Attribute Syntaxes

3990 An IPP object **MUST** be able to accept any of the attribute syntaxes defined in Section 4.1, including  
3991 their full range, in any operation in which a client may supply attributes or the system administrator may  
3992 configure attributes (by means outside the scope of this IPP/1.1 document). In particular for each  
3993 attribute that the IPP object supports whose attribute syntax is 'text', the IPP object **MUST** accept and  
3994 process both the 'textWithoutLanguage' and 'textWithLanguage' forms. Similarly, for each attribute that  
3995 the IPP object supports whose attribute syntax is 'name', the IPP object **MUST** accept and process both  
3996 the 'nameWithoutLanguage' and 'nameWithLanguage' forms. Furthermore, an IPP object **MUST** return  
3997 attributes to the client in operation responses that conform to the syntax specified in Section 4.1,  
3998 including their full range if supplied previously by a client.

#### 3999 5.3 Charset and Natural Language Requirements

4000 All clients and IPP objects **MUST** support the 'utf-8' charset as defined in section 4.1.7.

4001 IPP objects **MUST** be able to accept any client request which correctly uses the "attributes-natural-  
4002 language" operation attribute or the Natural Language Override mechanism on any individual attribute  
4003 whether or not the natural language is supported by the IPP object. If an IPP object supports a natural  
4004 language, then it **MUST** be able to translate (perhaps by table lookup) all generated 'text' or 'name'  
4005 attribute values into one of the supported languages (see section 3.1.4). That is, the IPP object that  
4006 supports a natural language **NEED NOT** be a general purpose translator of any arbitrary 'text' or 'name'  
4007 value supplied by the client into that natural language. However, the object **MUST** be able to translate  
4008 (automatically generate) any of its own attribute values and messages into that natural language.

## 4009 5.4 Security Conformance Requirements

4010 Conforming IPP Printer objects SHOULD support Transport Layer Security (TLS) protocol Version 1  
4011 (TLS) [RFC2246] access, MAY support access without TLS, or MAY support both means of access.

4012 Conforming IPP clients SHOULD support TLS access and non-TLS access. Note: This client  
4013 recommendation to support both means that conforming IPP clients will be able to inter-operate with any  
4014 IPP Printer object.

4015 For a detailed discussion of security considerations and the IPP application security profile required for  
4016 TLS support, see section 8.

## 4017 6. IANA Considerations (registered and private extensions)

4018 This section describes how IPP can be extended to allow the following registered and private extensions  
4019 to IPP:

- 4020 1. keyword attribute values
- 4021 2. enum attribute values
- 4022 3. attributes
- 4023 4. attribute syntaxes
- 4024 5. operations
- 4025 6. attribute groups
- 4026 7. status codes

4027

4028 Extensions registered for use with IPP/1.1 are OPTIONAL for client and IPP object conformance to the  
4029 IPP/1.1 Model specification.

4030 These extension procedures are aligned with the guidelines as set forth by the IESG [IANA-CON].  
4031 Section 12 describes how to propose new registrations for consideration. IANA will reject registration  
4032 proposals that leave out required information or do not follow the appropriate format described in  
4033 Section 12. IPP/1.1 may also be extended by an appropriate RFC that specifies any of the above  
4034 extensions.

### 4035 6.1 Typed 'keyword' and 'enum' Extensions

4036 IPP allows for 'keyword' and 'enum' extensions (see sections 4.1.2.3 and 4.1.4). This document uses  
4037 prefixes to the 'keyword' and 'enum' basic attribute syntax type in order to communicate extra  
4038 information to the reader through its name. This extra information is not represented in the protocol  
4039 because it is unimportant to a client or Printer object. The list below describes the prefixes and their  
4040 meaning.

4041 "type1": The IPP specification must be revised to add a new keyword or a new enum. No private  
4042 keywords or enums are allowed.

4043  
4044 "type2": Implementers can, at any time, add new keyword or enum values by proposing the  
4045 complete specification to IANA:

4046  
4047 iana@iana.org

4048  
4049 IANA will forward the registration proposal to the IPP Designated Expert who will review the  
4050 proposal with a mailing list that the Designated Expert keeps for this purpose. Initially, that list  
4051 will be the mailing list used by the IPP WG:

4052  
4053 ipp@pwg.org

4054  
4055 even after the IPP WG is disbanded as permitted by [IANA-CON]. The IPP Designated Expert  
4056 is appointed by the IESG Area Director responsible for IPP, according to [IANA-CON].

4057  
4058 When a type2 keyword or enum is approved, the IPP Designated Expert becomes the point of  
4059 contact for any future maintenance that might be required for that registration.

4060  
4061 "type3": Implementers can, at any time, add new keyword and enum values by submitting the  
4062 complete specification to IANA as for type2 who will forward the proposal to the IPP Designated  
4063 Expert. While no additional technical review is required, the IPP Designated Expert may, at  
4064 his/her discretion, forward the proposal to the same mailing list as for type2 registrations for  
4065 advice and comment.

4066  
4067 When a type3 keyword or enum is approved by the IPP Designated Expert, the original proposer  
4068 becomes the point of contact for any future maintenance that might be required for that  
4069 registration.

4070  
4071 For type2 and type3 keywords, the proposer includes the name of the keyword in the registration  
4072 proposal and the name is part of the technical review.

4073 After type2 and type3 enums specifications are approved, the IPP Designated Expert in consultation with  
4074 IANA assigns the next available enum number for each enum value.

4075 IANA will publish approved type2 and type3 keyword and enum attributes value registration  
4076 specifications in:

4077 ftp.isi.edu/iana/assignments/ipp/attribute-values/xxx/yyy.txt

4078 where xxx is the attribute name that specifies the initial values and yyy.txt is a descriptive file name that  
4079 contains one or more enums or keywords approved at the same time. For example, if several additional  
4080 enums for stapling are approved for use with the "finishings" attribute (and "finishings-default" and  
4081 "finishings-supported" attributes), IANA will publish the additional values in the file:

4082 ftp.isi.edu/iana/assignments/ipp/attribute-values/finishings/stapling.txt.

4083 Note: Some attributes are defined to be: 'type3 keywords' | 'name' which allows for attribute values to be  
4084 extended by a site administrator with administrator defined names. Such names are not registered with  
4085 IANA.

4086 By definition, each of the three types above assert some sort of registry or review process in order for  
4087 extensions to be considered valid. Each higher numbered level (1, 2, 3) tends to be decreasingly less  
4088 stringent than the previous level. Therefore, any typeN value MAY be registered using a process for  
4089 some typeM where M is less than N, however such registration is NOT REQUIRED. For example, a  
4090 type3 value MAY be registered in a type 1 manner (by being included in a future version of an IPP  
4091 specification), however, it is NOT REQUIRED.

4092 This specification defines keyword and enum values for all of the above types, including type3  
4093 keywords.

4094 For private (unregistered) keyword extensions, implementers SHOULD use keywords with a suitable  
4095 distinguishing prefix, such as "xxx-" where xxx is the (lowercase) fully qualified company name  
4096 registered with IANA for use in domain names [RFC1035]. For example, if the company XYZ Corp.  
4097 had obtained the domain name "XYZ.com", then a private keyword 'abc' would be: 'xyz.com-abc'.

4098 Note: RFC 1035 [RFC1035] indicates that while upper and lower case letters are allowed in domain  
4099 names, no significance is attached to the case. That is, two names with the same spelling but different  
4100 case are to be treated as if identical. Also, the labels in a domain name must follow the rules for  
4101 ARPANET host names: They must start with a letter, end with a letter or digit, and have as interior  
4102 characters only letters, digits, and hyphen. Labels must be 63 characters or less. Labels are separated by  
4103 the "." character.

4104 For private (unregistered) enum extension, implementers MUST use values in the reserved integer range  
4105 which is  $2^{*}30$  to  $2^{*}31-1$ .

## 4106 6.2 Attribute Extensibility

4107 Attribute names are type2 keywords. Therefore, new attributes may be registered and have the same  
4108 status as attributes in this document by following the type2 extension rules. For private (unregistered)  
4109 attribute extensions, implementers SHOULD use keywords with a suitable distinguishing prefix as  
4110 described in Section 6.1.

4111 IANA will publish approved attribute registration specifications as separate files:

4112 ftp.isi.edu/iana/assignments/ipp/attributes/xxx-yyy.txt

4113 where "xxx-yyy" is the new attribute name.

4114 If a new Printer object attribute is defined and its values can be affected by a specific document format,  
4115 its specification needs to contain the following sentence:

4116 "The value of this attribute returned in a Get-Printer-Attributes response MAY depend on the  
4117 "document-format" attribute supplied (see Section 3.2.5.1)."

4118 If the specification does not, then its value in the Get-Printer-Attributes response MUST NOT depend on  
4119 the "document-format" supplied in the request. When a new Job Template attribute is registered, the  
4120 value of the Printer attributes MAY vary with "document-format" supplied in the request without the  
4121 specification having to indicate so.

### 4122 6.3 Attribute Syntax Extensibility

4123 Attribute syntaxes are like type2 enums. Therefore, new attribute syntaxes may be registered and have  
4124 the same status as attribute syntaxes in this document by following the type2 extension rules described in  
4125 Section 6.1. The value codes that identify each of the attribute syntaxes are assigned in the "Encoding  
4126 and Transport" specification [IPP-PRO], including a designated range for private, experimental use.

4127 For attribute syntaxes, the IPP Designated Expert in consultation with IANA assigns the next attribute  
4128 syntax code in the appropriate range as specified in [IPP-PRO]. IANA will publish approved attribute  
4129 syntax registration specifications as separate files:

4130 ftp.isi.edu/iana/assignments/ipp/attribute-syntaxes/xxx-yyy.txt

4131 where 'xxx-yyy' is the new attribute syntax name.

### 4132 6.4 Operation Extensibility

4133 Operations may also be registered following the type2 procedures described in Section 6.1, though major  
4134 new operations will usually be done by a new standards track RFC that augments this document. For  
4135 private (unregistered) operation extensions, implementers MUST use the range for the "operation-id" in  
4136 requests specified in Section 4.4.13 "operations-supported" Printer attribute.

4137 For operations, the IPP Designated Expert in consultation with IANA assigns the next operation-id code  
4138 as specified in Section 4.4.13. IANA will publish approved operation registration specifications as  
4139 separate files:

4140 ftp.isi.edu/iana/assignments/ipp/operations/Xxx-Yyy.txt

4141 where "Xxx-Yyy" is the new operation name.

### 4142 6.5 Attribute Groups

4143 Attribute groups passed in requests and responses may be registered following the type2 procedures  
4144 described in Section 6.1. The tags that identify each of the attribute groups are assigned in [IPP-PRO].

4145 For attribute groups, the IPP Designated Expert in consultation with IANA assigns the next attribute  
4146 group tag code in the appropriate range as specified in [IPP-PRO]. IANA will publish approved  
4147 attribute group registration specifications as separate files:

4148 `ftp.isi.edu/iana/assignments/ipp/attribute-group-tags/xxx-yyy-tag.txt`

4149 where 'xxx-yyy-tag' is the new attribute group tag name.

## 4150 6.6 Status Code Extensibility

4151 Operation status codes may also be registered following the type2 procedures described in Section 6.1.  
4152 The values for status codes are allocated in ranges as specified in Section 14 for each status code class:

4153 "informational" - Request received, continuing process

4154 "successful" - The action was successfully received, understood, and accepted

4155 "redirection" - Further action must be taken in order to complete the request

4156 "client-error" - The request contains bad syntax or cannot be fulfilled

4157 "server-error" - The IPP object failed to fulfill an apparently valid request

4158

4159 For private (unregistered) operation status code extensions, implementers **MUST** use the top of each  
4160 range as specified in Section 14.

4161 For operation status codes, the IPP Designated Expert in consultation with IANA assigns the next status  
4162 code in the appropriate class range as specified in Section 14. IANA will publish approved status code  
4163 registration specifications as separate files:

4164 `ftp.isi.edu/iana/assignments/ipp/status-codes/xxx-yyy.txt`

4165 where "xxx-yyy" is the new operation status code keyword.

## 4166 6.7 Registration of MIME types/sub-types for document-formats

4167 The "document-format" attribute's syntax is 'mimeMediaType'. This means that valid values are Internet  
4168 Media Types (see Section 4.1.9). RFC 2045 [RFC2045] defines the syntax for valid Internet media  
4169 types. IANA is the registry for all Internet media types.

## 4170 6.8 Registration of charsets for use in 'charset' attribute values

4171 The "attributes-charset" attribute's syntax is 'charset'. This means that valid values are charsets names.  
4172 When a charset in the IANA registry has more than one name (alias), the name labeled as "(preferred  
4173 MIME name)", if present, **MUST** be used (see Section 4.1.7). IANA is the registry for charsets  
4174 following the procedures of [RFC2278].

## 4175 7. Internationalization Considerations

4176 Some of the attributes have values that are text strings and names which are intended for human  
4177 understanding rather than machine understanding (see the 'text' and 'name' attribute syntaxes in Sections  
4178 4.1.1 and 4.1.2).

4179 In each operation request, the client

- 4180 - identifies the charset and natural language of the request which affects each supplied 'text' and  
4181 'name' attribute value, and
  - 4182 - requests the charset and natural language for attributes returned by the IPP object in operation  
4183 responses (as described in Section 3.1.4.1).
- 4184

4185 In addition, the client MAY separately and individually identify the Natural Language Override of a  
4186 supplied 'text' or 'name' attribute using the 'textWithLanguage' and 'nameWithLanguage' technique  
4187 described section 4.1.1.2 and 4.1.2.2 respectively.

4188 All IPP objects MUST support the UTF-8 [RFC2279] charset in all 'text' and 'name' attributes supported.  
4189 If an IPP object supports more than the UTF-8 charset, the object MUST convert between them in order  
4190 to return the requested charset to the client according to Section 3.1.4.2. If an IPP object supports more  
4191 than one natural language, the object SHOULD return 'text' and 'name' values in the natural language  
4192 requested where those values are generated by the Printer (see Section 3.1.4.1).

4193 For Printers that support multiple charsets and/or multiple natural languages in 'text' and 'name'  
4194 attributes, different jobs may have been submitted in differing charsets and/or natural languages. All  
4195 responses MUST be returned in the charset requested by the client. However, the Get-Jobs operation  
4196 uses the 'textWithLanguage' and 'nameWithLanguage' mechanism to identify the differing natural  
4197 languages with each job attribute returned.

4198 The Printer object also has configured charset and natural language attributes. The client can query the  
4199 Printer object to determine the list of charsets and natural languages supported by the Printer object and  
4200 what the Printer object's configured values are. See the "charset-configured", "charset-supported",  
4201 "natural-language-configured", and "generated-natural-language-supported" Printer description attributes  
4202 for more details.

4203 The "charset-supported" attributed identifies the supported charsets. If a charset is supported, the IPP  
4204 object MUST be capable of converting to and from that charset into any other supported charset. In  
4205 many cases, an IPP object will support only one charset and it MUST be the UTF-8 charset.

4206 The "charset-configured" attribute identifies the one supported charset which is the native charset given  
4207 the current configuration of the IPP object (administrator defined).

4208 The "generated-natural-language-supported" attribute identifies the set of supported natural languages  
4209 for generated messages; it is not related to the set of natural languages that must be accepted for client  
4210 supplied 'text' and 'name' attributes. For client supplied 'text' and 'name' attributes, an IPP object MUST  
4211 accept ALL supplied natural languages. Just because a Printer object is currently configured to support

4212 'en-us' natural language does not mean that the Printer object should reject a job if the client supplies a  
4213 job name that is in 'fr-ca'.

4214 The "natural-language-configured" attribute identifies the one supported natural language for generated  
4215 messages which is the native natural language given the current configuration of the IPP object  
4216 (administrator defined).

4217 Attributes of type 'text' and 'name' are populated from different sources. These attributes can be  
4218 categorized into following groups (depending on the source of the attribute):

- 4219 1. Some attributes are supplied by the client (e.g., the client supplied "job-name", "document-name",  
4220 and "requesting-user-name" operation attributes along with the corresponding Job object's "job-  
4221 name" and "job-originating-user-name" attributes). The IPP object MUST accept these attributes  
4222 in any natural language no matter what the set of supported languages for generated messages
- 4223 2. Some attributes are supplied by the system administrator (e.g., the Printer object's "printer-name"  
4224 and "printer-location" attributes). These too can be in any natural language. If the natural  
4225 language for these attributes is different than what a client requests, then they must be reported  
4226 using the Natural Language Override mechanism.
- 4227 3. Some attributes are supplied by the device manufacturer (e.g., the Printer object's "printer-make-  
4228 and-model" attribute). These too can be in any natural language. If the natural language for  
4229 these attributes is different than what a client requests, then they must be reported using the  
4230 Natural Language Override mechanism.
- 4231 4. Some attributes are supplied by the operator (e.g., the Job object's "job-message-from-operator"  
4232 attribute). These too can be in any natural language. If the natural language for these attributes is  
4233 different than what a client requests, then they must be reported using the Natural Language  
4234 Override mechanism.
- 4235 5. Some attributes are generated by the IPP object (e.g., the Job object's "job-state-message"  
4236 attribute, the Printer object's "printer-state-message" attribute, and the "status-message" operation  
4237 attribute). These attributes can only be in one of the "generated-natural-language-supported"  
4238 natural languages. If a client requests some natural language for these attributes other than one of  
4239 the supported values, the IPP object SHOULD respond using the value of the "natural-language-  
4240 configured" attribute (using the Natural Language Override mechanism if needed).

4242 The 'text' and 'name' attributes specified in this version of this document (additional ones will be  
4243 registered according to the procedures in Section 6) are:

4244 Attributes	Source
4245 -----	-----
4246 Operation Attributes	
4247 job-name (name)	client
4248 document-name (name)	client
4249 requesting-user-name (name)	client
4250 status-message	Job or Printer object
4251	
4252 Job Template Attributes:	
4253 job-hold-until (keyword   name)	client matches administrator-configured

4254 job-hold-until-default (keyword | name) client matches administrator-configured  
 4255 job-hold-until-supported (keyword | name) client matches administrator-configured  
 4256 job-sheets (keyword | name) client matches administrator-configured  
 4257 job-sheets-default (keyword | name) client matches administrator-configured  
 4258 job-sheets-supported (keyword | name) client matches administrator-configured  
 4259 media (keyword | name) client matches administrator-configured  
 4260 media-default (keyword | name) client matches administrator-configured  
 4261 media-supported (keyword | name) client matches administrator-configured  
 4262 media-ready (keyword | name) client matches administrator-configured

4263

## 4264 Job Description Attributes:

4265 job-name (name) client or Printer object  
 4266 job-originating-user-name (name) Printer object  
 4267 job-state-message (text) Job or Printer object  
 4268 output-device-assigned (name(127)) administrator  
 4269 job-message-from-operator (text(127)) operator

4270

## 4271 Printer Description Attributes:

4272 printer-name (name(127)) administrator  
 4273 printer-location (text(127)) administrator  
 4274 printer-info (text(127)) administrator  
 4275 printer-make-and-model (text(127)) administrator or manufacturer  
 4276 printer-state-message (text) Printer object  
 4277 printer-message-from-operator (text(127)) operator

## 4278 8. Security Considerations

4279 IPP objects SHOULD be deployed over protocol stacks that support the Transport Layer Security (TLS)  
 4280 protocol [RFC2246]. Other IPP objects MAY be deployed over protocol stacks that do not support TLS.  
 4281 Some IPP objects MAY be deployed over both types of protocol stacks. Those IPP objects that support  
 4282 TLS, are capable of supporting mutual authentication as well as privacy of messages via multiple  
 4283 encryption schemes. An important point about security related information for TLS access to an IPP  
 4284 object, is that the security-related parameters (authentication, encryption keys, etc.) are "out-of-band" to  
 4285 the actual IPP protocol.

4286 An IPP object that does not support TLS MAY elect to support a transport layer that provides other  
 4287 security mechanisms. For example, in a mapping of IPP over HTTP/1.1 [IPP-PRO], if the IPP object  
 4288 does not support TLS, HTTP still allows for client authentication using Digest Access Authentication  
 4289 (DAA) [RFC2069].

4290 It is difficult to anticipate the security risks that might exist in any given IPP environment. For example,  
 4291 if IPP is used within a given corporation over a private network, the risks of exposing document data  
 4292 may be low enough that the corporation will choose not to use encryption on that data. However, if the  
 4293 connection between the client and the IPP object is over a public network, the client may wish to protect  
 4294 the content of the information during transmission through the network with encryption.

4295 Furthermore, the value of the information being printed may vary from one IPP environment to the next.  
4296 Printing payroll checks, for example, would have a different value than printing public information from  
4297 a file. There is also the possibility of denial-of-service attacks, but denial-of-service attacks against  
4298 printing resources are not well understood and there is no published precedents regarding this scenario.

4299 Once the authenticated identity of the requester has been supplied to the IPP object, the object uses that  
4300 identity to enforce any authorization policy that might be in place. For example, one site's policy might  
4301 be that only the job owner is allowed to cancel a job. The details and mechanisms to set up a particular  
4302 access control policy are not part of IPP/1.1, and must be established via some other type of  
4303 administrative or access control framework. However, there are operation status codes that allow an IPP  
4304 server to return information back to a client about any potential access control violations for an IPP  
4305 object.

4306 During a create operation, the client's identity is recorded in the Job object in an implementation-defined  
4307 attribute. This information can be used to verify a client's identity for subsequent operations on that Job  
4308 object in order to enforce any access control policy that might be in effect. See section 8.3 below for  
4309 more details.

4310 Since the security levels or the specific threats that any given IPP system administrator may be  
4311 concerned with cannot be anticipated, IPP MUST be capable of operating with different security  
4312 mechanisms and security policies as required by the individual installation. Security policies might vary  
4313 from very strong, to very weak, to none at all, and corresponding security mechanisms will be required.  
4314 TLS supports the type of negotiated levels of security required by most, if not all, potential IPP  
4315 environments. IPP environments that require no security can elect to deploy IPP objects that do not  
4316 utilize the optional TLS security mechanisms.

## 4317 8.1 Security Scenarios

4318 The following sections describe specific security attacks for IPP environments. Where examples are  
4319 provided they should be considered illustrative of the environment and not an exhaustive set. Not all of  
4320 these environments will necessarily be addressed in initial implementations of IPP.

### 4321 8.1.1 Client and Server in the Same Security Domain

4322 This environment is typical of internal networks where traditional office workers print the output of  
4323 personal productivity applications on shared work-group printers, or where batch applications print their  
4324 output on large production printers. Although the identity of the user may be trusted in this environment,  
4325 a user might want to protect the content of a document against such attacks as eavesdropping, replaying  
4326 or tampering.

### 4327 8.1.2 Client and Server in Different Security Domains

4328 Examples of this environment include printing a document created by the client on a publicly available  
4329 printer, such as at a commercial print shop; or printing a document remotely on a business associate's  
4330 printer. This latter operation is functionally equivalent to sending the document to the business associate

4331 as a facsimile. Printing sensitive information on a Printer in a different security domain requires strong  
4332 security measures. In this environment authentication of the printer is required as well as protection  
4333 against unauthorized use of print resources. Since the document crosses security domains, protection  
4334 against eavesdropping and document tampering are also required. It will also be important in this  
4335 environment to protect Printers against "spamming" and malicious document content.

### 4336 8.1.3 Print by Reference

4337 When the document is not stored on the client, printing can be done by reference. That is, the print  
4338 request can contain a reference, or pointer, to the document instead of the actual document itself.  
4339 Standard methods currently do not exist for remote entities to "assume" the credentials of a client for  
4340 forwarding requests to a 3rd party. It is anticipated that Print-By-Reference will be used to access  
4341 "public" documents and that sophisticated methods for authenticating "proxies" will not be specified for  
4342 version 1 of IPP.

## 4343 8.2 URIs for TLS and non-TLS Access

4344 As described earlier, an IPP object SHOULD support TLS access, MAY non-TLS access, or both. The  
4345 "printer-uri-supported" attribute contains the Printer object's URI(s). Its companion attribute, "uri-  
4346 security-supported", identifies the security mechanism used for each URI listed in the "printer-uri-  
4347 supported" attribute. For each Printer operation request, a client MUST supply only one URI in the  
4348 "printer-uri" operation attribute. In other words, even though the Printer supports more than one URI,  
4349 the client only interacts with the Printer object using one of its URIs. This duality is not needed for Job  
4350 objects, since the Printer object is the factory for Job objects, and the Printer object will generate the  
4351 correct URI for new Job objects depending on the Printer object's security configuration.

### 4352 8.3 The "requesting-user-name" (name(MAX)) Operation Attribute

4353 Each operation MUST specify the user who is performing the operation in both of the following two  
4354 ways:

- 4355 1) via the REQUIRED "requesting-user-name" operation attribute that a client SHOULD supply in  
4356 all operations. The client MUST obtain the value for this attribute from an environmental or  
4357 network login name for the user, rather than allowing the user to supply any value. If the client  
4358 does not supply a value for "requesting-user-name", the printer MUST assume that the client is  
4359 supplying some anonymous name, such as "anonymous".
- 4360 2) via an authentication mechanism of the underlying transport which may be configured to give no  
4361 authentication information.

4362  
4363 There are six cases to consider:

- 4364 a) the authentication mechanism gives no information, and the client doesn't specify "requesting-  
4365 user-name".
- 4366 b) the authentication mechanism gives no information, but the client specifies "requesting-user-  
4367 name".

- 4368 c) the authentication mechanism specifies a user which has no human readable representation, and  
4369 the client doesn't specify "requesting-user-name".
- 4370 d) the authentication mechanism specifies a user which has no human readable representation, but  
4371 the client specifies "requesting-user-name".
- 4372 e) the authentication mechanism specifies a user which has a human readable representation. The  
4373 Printer object ignores the "requesting-user-name".
- 4374 f) the authentication mechanism specifies a user who is trusted and whose name means that the  
4375 value of the "requesting-user-name", which **MUST** be present, is treated as the authenticated  
4376 name.  
4377

4378 Note: Case "f" is intended for a tightly coupled gateway and server to work together so that the "user"  
4379 name is able to be that of the gateway client and not that of the gateway. Because most, if not all, system  
4380 vendors will initially implement IPP via a gateway into their existing print system, this mechanism is  
4381 necessary unless the authentication mechanism allows a gateway (client) to act on behalf of some other  
4382 client.

4383 The user-name has two forms:

- 4384 - one that is human readable: it is held in the **REQUIRED** "job-originating-user-name" Job  
4385 Description attribute which is set during the job creation operations. It is used for presentation  
4386 only, such as returning in queries or printing on start sheets
- 4387 - one for authorization: it is held in an undefined (by IPP) Job object attribute which is set by the job  
4388 creation operation. It is used to authorize other operations, such as Send-Document, Send-URI,  
4389 Cancel-Job, to determine the user when the "my-jobs" attribute is specified with Get-Jobs, and to  
4390 limit what attributes and values to return with Get-Job-Attributes and Get-Jobs.  
4391

4392 The human readable user name:

- 4393 - is the value of the "requesting-user-name" for cases b, d and f.  
4394 - comes from the authentication mechanism for case e  
4395 - is some anonymous name, such as "anonymous" for cases a and c.  
4396

4397 The user name used for authorization:

- 4398 - is the value of the "requesting-user-name" for cases b and f.  
4399 - comes from the authentication mechanism for cases c, d and e  
4400 - is some anonymous name, such as "anonymous" for case a.  
4401

4402 The essence of these rules for resolving conflicting sources of user-names is that a printer  
4403 implementation is free to pick either source as long as it achieves consistent results. That is, if a user  
4404 uses the same path for a series of requests, the requests **MUST** appear to come from the same user from  
4405 the standpoint of both the human-readable user name and the user name for authorization. This rule  
4406 **MUST** continue to apply even if a request could be authenticated by two or more mechanisms. It doesn't  
4407 matter which of several authentication mechanisms a Printer uses as long as it achieves consistent

4408 results. If a client uses more than one authentication mechanism, it is recommended that an  
4409 administrator make all credentials resolve to the same user and user-name as much as possible.

#### 4410 8.4 Restricted Queries

4411 In many IPP operations, a client supplies a list of attributes to be returned in the response. For security  
4412 reasons, an IPP object may be configured not to return all attributes (or all values) that a client requests.  
4413 The job attributes returned MAY depend on whether the requesting user is the same as the user that  
4414 submitted the job. The IPP object MAY even return none of the requested attributes. In such cases, the  
4415 status returned is the same as if the object had returned all requested attributes. The client cannot tell by  
4416 such a response whether the requested attribute was present or absent on the object.

#### 4417 8.5 Operations performed by operators and system administrators

4418 For the three printer operations Pause-Printer, Resume-Printer, and Purge-Jobs (see sections 3.2.7, 3.2.8  
4419 and 3.2.9), the requesting user is intended to be an operator or administrator of the Printer object (see  
4420 section 1). The means for authorizing an operator or administrator of the Printer object are not specified  
4421 in this document.

#### 4422 8.6 Queries on jobs submitted using non-IPP protocols

4423 If the device that an IPP Printer is representing is able to accept jobs using other job submission  
4424 protocols in addition to IPP, it is RECOMMENDED that such an implementation at least allow such  
4425 "foreign" jobs to be queried using Get-Jobs returning "job-id" and "job-uri" as 'unknown'. Such an  
4426 implementation NEED NOT support all of the same IPP job attributes as for IPP jobs. The IPP object  
4427 returns the 'unknown' out-of-band value for any requested attribute of a foreign job that is supported for  
4428 IPP jobs, but not for foreign jobs.

4429 It is further RECOMMENDED, that the IPP Printer generate "job-id" and "job-uri" values for such  
4430 "foreign jobs", if possible, so that they may be targets of other IPP operations, such as Get-Job-Attributes  
4431 and Cancel-Job. Such an implementation also needs to deal with the problem of authentication of such  
4432 foreign jobs. One approach would be to treat all such foreign jobs as belonging to users other than the  
4433 user of the IPP client. Another approach would be for the foreign job to belong to 'anonymous'. Only if  
4434 the IPP client has been authenticated as an operator or administrator of the IPP Printer object, could the  
4435 foreign jobs be queried by an IPP request. Alternatively, if the security policy is to allow users to query  
4436 other users' jobs, then the foreign jobs would also be visible to an end-user IPP client using Get-Jobs and  
4437 Get-Job-Attributes.

#### 4438 8.7 IPP Security Application Profile for TLS

4439 The IPP application profile for TLS follows the standard "Mandatory Cipher Suites" requirement  
4440 as documented in the TLS specification [RFC2246].

- 4441 If a conforming IPP object supports TLS, it MUST implement and support the "Mandatory Cipher  
4442 Suites" as specified in the TLS specification [RFC2246] and MAY support additional cipher suites.
- 4443 A conforming IPP client SHOULD support TLS including the "Mandatory Cipher Suites" as specified in  
4444 the TLS specification [RFC2246]. A conforming IPP client MAY support additional cipher suites.  
4445 Client implementations MUST NOT assume any other cipher suites are supported by an IPP Printer  
4446 object.
- 4447 9. See the TLS specification [RFC2246] for a discussion of any government export restrictions on  
4448 implementations conforming to the "Mandatory Cipher Suites". References
- 4449 [ASCII]  
4450 Coded Character Set - 7-bit American Standard Code for Information Interchange (ASCII), ANSI  
4451 X3.4-1986. This standard is the specification of the US-ASCII charset.
- 4452 [BCP-11]  
4453 [Bradner S.](#), [Hovey R.](#), "The Organizations Involved in the IETF Standards Process", 1996/10/29  
4454 (RFC 2028)
- 4455 [HTPP]  
4456 J. Barnett, K. Carter, R. DeBry, "Initial Draft - Hypertext Printing Protocol - HTPP/1.0",  
4457 October 1996, <ftp://ftp.pwg.org/pub/pwg/ipp/historic/http/overview.ps.gz>
- 4458 [IANA-CON]  
4459 Narte, T. and Alvestrand, H.T.: Guidelines for Writing an IANA Considerations Section in  
4460 RFCs, Work in Progress, draft-iesg-iana-considerations-04.txt, May 21, 1998.
- 4461 [IANA-CS]  
4462 IANA Registry of Coded Character Sets: <ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets>
- 4463 [IANA-MT]  
4464 IANA Registry of Media Types: <ftp://ftp.isi.edu/in-notes/iana/assignments/media-types/>
- 4465 [IPP-IIG]  
4466 Hastings, T., Manros, C., "Internet Printing Protocol/1.1: draft-ietf-ipp-implementers-guide-v11-  
4467 ???.txt, ?? 1999, work in progress.
- 4468 [IPP-IIG1.0]  
4469 Hastings, T., Manros, C., "Internet Printing Protocol/1.0: Implementer's Guide", draft-ietf-ipp-  
4470 implementers-guide-01.txt, February 1999, work in progress.
- 4471 [IPP LPD]  
4472 Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols",  
4473 draft-ietf-ipp-lpd-ipp-map-05.txt, November 1998.

- 4474 [IPP-MOD1.0]  
4475 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model  
4476 and Semantics", draft-ietf-ipp-model-11.txt, November, 1998.
- 4477 [IPP-PRO]  
4478 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.1: Encoding and  
4479 Transport", draft-ietf-ipp-protocol-v11-00.txt, February, 1999.
- 4480 [IPP-PRO1.0]  
4481 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.0: Encoding and  
4482 Transport", draft-ietf-ipp-protocol-07.txt, November, 1998.
- 4483 [IPP-RAT]  
4484 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",  
4485 draft-ietf-ipp-rat-04.txt, November, 1998.
- 4486 [IPP-REQ]  
4487 Wright, D., "Design Goals for an Internet Printing Protocol", draft-ietf-ipp-req-03.txt, November,  
4488 1998.
- 4489 [ISO10646-1]  
4490 ISO/IEC 10646-1:1993, "Information technology -- Universal Multiple-Octet Coded Character  
4491 Set (UCS) - Part 1: Architecture and Basic Multilingual Plane, JTC1/SC2."
- 4492 [ISO8859-1]  
4493 ISO/IEC 8859-1:1987, "Information technology -- 8-bit One-Byte Coded Character Set - Part 1:  
4494 Latin Alphabet Nr 1", 1987, JTC1/SC2.
- 4495 [ISO10175]  
4496 ISO/IEC 10175 Document Printing Application (DPA), June 1996.
- 4497 [LDPA]  
4498 T. Hastings, S. Isaacson, M. MacKay, C. Manros, D. Taylor, P. Zehler, "LDPA - Lightweight  
4499 Document Printing Application", October 1996,  
4500 <ftp://ftp.pwg.org/pub/pwg/ipp/historic/ldpa/ldpa8.pdf.gz>
- 4501 [P1387.4]  
4502 Kirk, M. (editor), POSIX System Administration - Part 4: Printing Interfaces, POSIX 1387.4 D8,  
4503 1994.
- 4504 [PSIS] Herriot, R. (editor), X/Open A Printing System Interoperability Specification (PSIS), August  
4505 1995.
- 4506 [PWG]  
4507 Printer Working Group, <http://www.pwg.org>.

- 4508 [RFC1035]  
4509 P. Mockapetris, "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION", RFC  
4510 1035, November 1987.
- 4511 [RFC1179]  
4512 McLaughlin, L. III, (editor), "Line Printer Daemon Protocol" RFC 1179, August 1990.
- 4513 [RFC1759]  
4514 Smith, R., Wright, F., Hastings, T., Zilles, S., and Gyllenskog, J., "Printer MIB", RFC 1759,  
4515 March 1995.
- 4516 [RFC1766]  
4517 H. Alvestrand, "Tags for the Identification of Languages", RFC 1766, March 1995.
- 4518 [RFC1903]  
4519 J. Case, et.al., "Textual Conventions for Version 2 of the Simple Network Management Protocol  
4520 (SNMP v2)" RFC 1903, January 1996.
- 4521 [RFC1952]  
4522 P. Deutsch, "GZIP file format specification version 4.3", RFC 1952, May 1996.
- 4523 [RFC2026]  
4524 S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.
- 4525 [RFC2045]  
4526 N. Fried, N. Borenstein, ", Multipurpose Internet Mail Extensions (MIME) Part One: Format of  
4527 Internet Message Bodies " RFC 2045, November 1996.
- 4528 [RFC2046]  
4529 Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. N. Freed & N.  
4530 Borenstein. November 1996. (Obsoletes RFC1521, RFC1522, RFC1590), RFC 2046.
- 4531 [RFC2048]  
4532 N. Freed, J. Klensin & J. Postel, "Multipurpose Internet Mail Extension (MIME) Part Four:  
4533 Registration Procedures". RFC 2048, November 1996.
- 4534 [RFC2068]  
4535 R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, "Hypertext Transfer Protocol -  
4536 HTTP/1.1", RFC 2068, January 1997
- 4537 [RFC2069]  
4538 J. Franks, P. Hallam-Baker, J. Hostetler, P. Leach, A. Luotonen, E. Sink, L. Stewart, "An  
4539 Extension to HTTP: Digest Access Authentication", RFC-2069, Jan 1997.
- 4540 [RFC2119]  
4541 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 , March  
4542 1997

- 4543 [RFC2228]  
4544 M. Horowitz, S. Lunt, "FTP Security Extensions", RFC 2228, October 1997.
- 4545 [RFC2246]  
4546 T. Dierks, C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.
- 4547 [RFC2277]  
4548 H. Alvestrand, "IETF Policy on Character Sets and Languages" RFC 2277, January 1998.
- 4549 [RFC2278]  
4550 N. Freed, J. Postel: "IANA CharSet Registration Procedures", RFC 2278, January 1998.
- 4551 [RFC2279]  
4552 F. Yergeau , "UTF-8, a transformation format of ISO 10646", RFC 2279. January 1998.
- 4553 [RFC2316]  
4554 S. Bellovin , "Report of the IAB Security Architecture Workshop", RFC 2316, April 1998.
- 4555 [RFC2396]  
4556 Berners-Lee, T., Fielding, R., Masinter, L., "Uniform Resource Identifiers (URI): Generic  
4557 Syntax", RFC 2396, August 1998.
- 4558 [SSL]  
4559 Netscape, The SSL Protocol, Version 3, (Text version 3.02), November 1996.
- 4560 [SWP]  
4561 P. Moore, B. Jahromi, S. Butler, "Simple Web Printing SWP/1.0", May 7, 1997,  
4562 [ftp://ftp.pwg.org/pub/pwg/ipp/new\\_PRO/swp9705.pdf](ftp://ftp.pwg.org/pub/pwg/ipp/new_PRO/swp9705.pdf)

4563 10. Notices

4564 The IETF takes no position regarding the validity or scope of any intellectual property or other rights that  
4565 might be claimed to pertain to the implementation or use of the technology described in this document  
4566 or the extent to which any license under such rights might or might not be available; neither does it  
4567 represent that it has made any effort to identify any such rights. Information on the IETF's procedures  
4568 with respect to rights in standards-track and standards-related documentation can be found in BCP-  
4569 11[BCP-11]. Copies of claims of rights made available for publication and any assurances of licenses to  
4570 be made available, or the result of an attempt made to obtain a general license or permission for the use  
4571 of such proprietary rights by implementers or users of this specification can be obtained from the IETF  
4572 Secretariat.

4573 The IETF invites any interested party to bring to its attention any copyrights, patents or patent  
4574 applications, or other proprietary rights which may cover technology that may be required to practice this  
4575 standard. Please address the information to the IETF Executive Director.

4576 Copyright (C) The Internet Society (1999). All Rights Reserved.

4577 This document and translations of it may be copied and furnished to others, and derivative works that  
4578 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published  
4579 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright  
4580 notice and this paragraph are included on all such copies and derivative works. However, this document  
4581 itself may not be modified in any way, such as by removing the copyright notice or references to the  
4582 Internet Society or other Internet organizations, except as needed for the purpose of developing Internet  
4583 standards in which case the procedures for copyrights defined in the Internet Standards process must be  
4584 followed, or as required to translate it into languages other than English.

4585 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or  
4586 its successors or assigns.

4587 This document and the information contained herein is provided on an "AS IS" basis and THE  
4588 INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL  
4589 WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY  
4590 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY  
4591 RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A  
4592 PARTICULAR PURPOSE.

## 4593 11. Author's Address

4594 Scott A. Isaacson (Editor)  
4595 Novell, Inc.  
4596 122 E 1700 S  
4597 Provo, UT 84606  
4598  
4599 Phone: 801-861-7366  
4600 Fax: 801-861-2517  
4601 e-mail: [sisaacson@novell.com](mailto:sisaacson@novell.com)

4602  
4603 Tom Hastings  
4604 Xerox Corporation  
4605 737 Hawaii St. ESAE 231  
4606 El Segundo, CA 90245  
4607  
4608 Phone: 310-333-6413  
4609 Fax: 310-333-5514  
4610 e-mail: [hastings@cp10.es.xerox.com](mailto:hastings@cp10.es.xerox.com)

4611  
4612 Robert Herriot  
4613 Xerox Corp.  
4614 3400 Hill View Ave, Building 1  
4615 Palo Alto, CA 94304

4616  
4617 Phone: 650-813-7696  
4618 Fax: 650-813-6860  
4619 e-mail: robert.herriot@pahv.xerox.com

4620  
4621 Roger deBry  
4622 HUC/003G  
4623 IBM Corporation  
4624 P.O. Box 1900  
4625 Boulder, CO 80301-9191

4626  
4627 Phone: (303) 924-4080  
4628 Fax: (303) 924-9889  
4629 e-mail: debry@vnet.ibm.com

4630  
4631 Patrick Powell  
4632 Astart Technologies  
4633 9475 Chesapeake Dr., Suite D  
4634 San Diego, CA 95123

4635  
4636 Phone: (619) 874-6543  
4637 Fax: (619) 279-8424  
4638 e-mail: papowell@astart.com

4639  
4640 IPP Mailing List: [ipp@pwg.org](mailto:ipp@pwg.org)  
4641 IPP Mailing List Subscription: [ipp-request@pwg.org](mailto:ipp-request@pwg.org)  
4642 IPP Web Page: <http://www.pwg.org/ipp/>  
4643

4644 Implementers of this specification are encouraged to join IPP Mailing List in order to participate in any  
4645 discussions of clarification issues and review of registration proposals for additional attributes and  
4646 values.

4647  
4648 Other Participants:

4649 Chuck Adams - Tektronix  
4650 Jeff Barnett - IBM  
4651 Ron Bergman - Dataproducts Corp.  
4652 Sylvan Butler - HP  
4653 Keith Carter - IBM Corporation  
4654 Jeff Copeland - QMS  
4655 Andy Davidson - Tektronix  
4656 Mabry Dozier - QMS  
4657 Lee Farrell - Canon Information Systems  
4658 Steve Gebert - IBM  
4659 Babek Jahromi - Microsoft  
4660 David Kellerman - Northlake Software

4661 Rick Landau - Digital  
4662 Greg LeClair - Epson  
4663 Harry Lewis - IBM  
4664 Pete Loya - HP  
4665 Ray Lutz - Cognisys  
4666 Mike MacKay - Novell, Inc.  
4667 Daniel Manchala - Xerox  
4668 Carl-Uno Manros - Xerox  
4669 Jay Martin - Underscore  
4670 Larry Masinter - Xerox  
4671 Stan McConnell - Xerox  
4672 Ira McDonald - High North Inc.  
4673 Paul Moore - Microsoft  
4674 Tetsuya Morita - Ricoh  
4675 Yuichi Niwa - Ricoh  
4676 Pat Nogay - IBM  
4677 Ron Norton - Printronics  
4678 Bob Pentecost - HP  
4679 Rob Rhoads - Intel  
4680 Xavier Riley - Xerox  
4681 David Roach - Unisys  
4682 Stuart Rowley - Kyocera  
4683 Hiroyuki Sato - Canon  
4684 Bob Setterbo - Adobe  
4685 Devon Taylor - Novell, Inc.  
4686 Mike Timperman - Lexmark  
4687 Randy Turner - Sharp  
4688 Atsushi Yuki - Kyocera  
4689 Rick Yardumian - Xerox  
4690 Lloyd Young - Lexmark  
4691 Bill Wagner - DPI  
4692 Jim Walker - DAZEL  
4693 Chris Wellens - Interworking Labs  
4694 Rob Whittle - Novell, Inc.  
4695 Don Wright - Lexmark  
4696 Peter Zehler - Xerox  
4697 Steve Zilles - Adobe

## 4698 12. Formats for IPP Registration Proposals

4699 In order to propose an IPP extension for registration, the proposer must submit an application to IANA  
4700 by email to "iana@iana.org" or by filling out the appropriate form on the IANA web pages  
4701 (<http://www.iana.org>). This section specifies the required information and the formats for proposing  
4702 registrations of extensions to IPP as provided in Section 6 for:

4703

4704 1. type2 'keyword' attribute values

4705 2. type3 'keyword' attribute values

4706 3. type2 'enum' attribute values

4707 4. type3 'enum' attribute values

4708 5. attributes

4709 6. attribute syntaxes

4710 7. operations

4711 8. status codes

## 4712 12.1 Type2 keyword attribute values registration

4713 Type of registration: type2 keyword attribute value

4714 Name of attribute to which this keyword specification is to be added:

4715 Proposed keyword name of this keyword value:

4716 Specification of this keyword value (follow the style of IPP Model Section 4.1.2.3):

4717 Name of proposer:

4718 Address of proposer:

4719 Email address of proposer:

4720

4721 Note: For type2 keywords, the Designated Expert will be the point of contact for the approved  
4722 registration specification, if any maintenance of the registration specification is needed.

## 4723 12.2 Type3 keyword attribute values registration

4724 Type of registration: type3 keyword attribute value

4725 Name of attribute to which this keyword specification is to be added:

4726 Proposed keyword name of this keyword value:

4727 Specification of this keyword value (follow the style of IPP Model Section 4.1.2.3):

4728 Name of proposer:

4729 Address of proposer:

4730 Email address of proposer:

4731

4732 Note: For type3 keywords, the proposer will be the point of contact for the approved registration  
4733 specification, if any maintenance of the registration specification is needed.

## 4734 12.3 Type2 enum attribute values registration

4735 Type of registration: type2 enum attribute value

4736 Name of attribute to which this enum specification is to be added:  
4737 Keyword symbolic name of this enum value:  
4738 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):  
4739 Specification of this enum value (follow the style of IPP Model Section 4.1.4):  
4740 Name of proposer:  
4741 Address of proposer:  
4742 Email address of proposer:  
4743  
4744 Note: For type2 enums, the Designated Expert will be the point of contact for the approved registration  
4745 specification, if any maintenance of the registration specification is needed.

#### 4746 12.4 Type3 enum attribute values registration

4747 Type of registration: type3 enum attribute value  
4748 Name of attribute to which this enum specification is to be added:  
4749 Keyword symbolic name of this enum value:  
4750 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):  
4751 Specification of this enum value (follow the style of IPP Model Section 4.1.4):  
4752 Name of proposer:  
4753 Address of proposer:  
4754 Email address of proposer:  
4755  
4756 Note: For type3 enums, the proposer will be the point of contact for the approved registration  
4757 specification, if any maintenance of the registration specification is needed.

#### 4758 12.5 Attribute registration

4759 Type of registration: attribute  
4760 Proposed keyword name of this attribute:  
4761 Types of attribute (Operation, Job Template, Job Description, Printer Description):  
4762 Operations to be used with if the attribute is an operation attribute:  
4763 Object (Job, Printer, etc. if bound to an object):  
4764 Attribute syntax(es) (include 1setOf and range as in Section 4.2):  
4765 If attribute syntax is 'keyword' or 'enum', is it type2 or type3:  
4766 If this is a Printer attribute, MAY the value returned depend on "document-format" (See Section 6.2):  
4767 If this is a Job Template attribute, how does its specification depend on the value of the "multiple-  
4768 document-handling" attribute:  
4769 Specification of this attribute (follow the style of IPP Model Section 4.2):  
4770 Name of proposer:  
4771 Address of proposer:  
4772 Email address of proposer:  
4773  
4774 Note: For attributes, the IPP Designated Expert will be the point of contact for the approved registration  
4775 specification, if any maintenance of the registration specification is needed.

## 4776 12.6 Attribute Syntax registration

4777 Type of registration: attribute syntax

4778 Proposed name of this attribute syntax:

4779 Type of attribute syntax (integer, octetString, character-string, see [IPP-PRO]):

4780 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

4781 Specification of this attribute (follow the style of IPP Model Section 4.1):

4782 Name of proposer:

4783 Address of proposer:

4784 Email address of proposer:

4785

4786 Note: For attribute syntaxes, the IPP Designated Expert will be the point of contact for the approved  
4787 registration specification, if any maintenance of the registration specification is needed.

## 4788 12.7 Operation registration

4789 Type of registration: operation

4790 Proposed name of this operation:

4791 Numeric operation-id value (to be assigned by the IPP Designated Expert in consultation with IANA):

4792 Object Target (Job, Printer, etc. that operation is upon):

4793 Specification of this attribute (follow the style of IPP Model Section 3):

4794 Name of proposer:

4795 Address of proposer:

4796 Email address of proposer:

4797

4798 Note: For operations, the IPP Designated Expert will be the point of contact for the approved  
4799 registration specification, if any maintenance of the registration specification is needed.

## 4800 12.8 Attribute Group registration

4801 Type of registration: attribute group

4802 Proposed name of this attribute group:

4803 Numeric tag according to [IPP-PRO] (to be assigned by the IPP Designated Expert in consultation with  
4804 IANA):

4805 Operation requests and group number for each operation in which the attribute group occurs:

4806 Operation responses and group number for each operation in which the attribute group occurs:

4807 Specification of this attribute group (follow the style of IPP Model Section 3):

4808 Name of proposer:

4809 Address of proposer:

4810 Email address of proposer:

4811

4812 Note: For attribute groups, the IPP Designated Expert will be the point of contact for the approved  
4813 registration specification, if any maintenance of the registration specification is needed.

## 4814 12.9 Status code registration

4815 Type of registration: status code

4816 Keyword symbolic name of this status code value:

4817 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

4818 Operations that this status code may be used with:

4819 Specification of this status code (follow the style of IPP Model Section 14 APPENDIX B: Status Codes  
4820 and Suggested Status Code Messages):

4821 Name of proposer:

4822 Address of proposer:

4823 Email address of proposer:

4824

4825 Note: For status codes, the Designated Expert will be the point of contact for the approved registration  
4826 specification, if any maintenance of the registration specification is needed.

## 4827 13. APPENDIX A: Terminology

4828 This specification uses the terminology defined in this section.

## 4829 13.1 Conformance Terminology

4830 The key words "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT",

4831 "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in

4832 RFC 2119 [RFC2119].

## 4833 13.1.1 NEED NOT

4834 This term is not included in RFC 2119. The verb "NEED NOT" indicates an action that the subject of

4835 the sentence does not have to implement in order to claim conformance to the standard. The verb

4836 "NEED NOT" is used instead of "MAY NOT" since "MAY NOT" sounds like a prohibition.

## 4837 13.2 Model Terminology

## 4838 13.2.1 Keyword

4839 Keywords are used within this document as identifiers of semantic entities within the abstract model (see

4840 section 4.1.2.3). Attribute names, some attribute values, attribute syntaxes, and attribute group names

4841 are represented as keywords.

## 4842 13.2.2 Attributes

4843 An attribute is an item of information that is associated with an instance of an IPP object. An attribute

4844 consists of an attribute name and one or more attribute values. Each attribute has a specific attribute

4845 syntax. All object attributes are defined in section 3.3.5 and all operation attributes are defined in  
4846 section 3.

4847 Job Template Attributes are described in section 4.2. The client optionally supplies Job Template  
4848 attributes in a create request (operation requests that create Job objects). The Printer object has  
4849 associated attributes which define supported and default values for the Printer.

#### 4850 13.2.2.1 Attribute Name

4851 Each attribute is uniquely identified in this document by its attribute name. An attribute name is a  
4852 keyword. The keyword attribute name is given in the section header describing that attribute. In running  
4853 text in this document, attribute names are indicated inside double quotation marks ("") where the  
4854 quotation marks are not part of the keyword itself.

#### 4855 13.2.2.2 Attribute Group Name

4856 Related attributes are grouped into named groups. The name of the group is a keyword. The group  
4857 name may be used in place of naming all the attributes in the group explicitly. Attribute groups are  
4858 defined in section 3.

#### 4859 13.2.2.3 Attribute Value

4860 Each attribute has one or more values. Attribute values are represented in the syntax type specified for  
4861 that attribute. In running text in this document, attribute values are indicated inside single quotation  
4862 marks ('), whether their attribute syntax is keyword, integer, text, etc. where the quotation marks are not  
4863 part of the value itself.

#### 4864 13.2.2.4 Attribute Syntax

4865 Each attribute is defined using an explicit syntax type. In this document, each syntax type is defined as a  
4866 keyword with specific meaning. The "Encoding and Transport" document [IPP-PRO] indicates the  
4867 actual "on-the-wire" encoding rules for each syntax type. Attribute syntax types are defined in section  
4868 4.1.

#### 4869 13.2.3 Supports

4870 By definition, a Printer object supports an attribute only if that Printer object responds with the  
4871 corresponding attribute populated with some value(s) in a response to a query for that attribute. A  
4872 Printer object supports an attribute value if the value is one of the Printer object's "supported values"  
4873 attributes. The device behind a Printer object may exhibit a behavior that corresponds to some IPP  
4874 attribute, but if the Printer object, when queried for that attribute, doesn't respond with the attribute, then  
4875 as far as IPP is concerned, that implementation does not support that feature. If the Printer object's "xxx-  
4876 supported" attribute is not populated with a particular value (even if that value is a legal value for that  
4877 attribute), then that Printer object does not support that particular value.

4878 A conforming implementation **MUST** support all **REQUIRED** attributes. However, even for  
4879 **REQUIRED** attributes, conformance to IPP does not mandate that all implementations support all  
4880 possible values representing all possible job processing behaviors and features. For example, if a given  
4881 instance of a Printer supports only certain document formats, then that Printer responds with the  
4882 "document-format-supported" attribute populated with a set of values, possibly only one, taken from the  
4883 entire set of possible values defined for that attribute. This limited set of values represents the Printer's  
4884 set of supported document formats. Supporting an attribute and some set of values for that attribute  
4885 enables IPP end users to be aware of and make use of those features associated with that attribute and  
4886 those values. If an implementation chooses to not support an attribute or some specific value, then IPP  
4887 end users would have no ability to make use of that feature within the context of IPP itself. However,  
4888 due to existing practice and legacy systems which are not IPP aware, there might be some other  
4889 mechanism outside the scope of IPP to control or request the "unsupported" feature (such as embedded  
4890 instructions within the document data itself).

4891 For example, consider the "finishings-supported" attribute.

- 4892 1) If a Printer object is not physically capable of stapling, the "finishings-supported" attribute **MUST**  
4893 **NOT** be populated with the value of 'staple'.
- 4894 2) A Printer object is physically capable of stapling, however an implementation chooses not to  
4895 support stapling in the IPP "finishings" attribute. In this case, 'staple' **MUST NOT** be a value in  
4896 the "finishings-supported" Printer object attribute. Without support for the value 'staple', an IPP  
4897 end user would have no means within the protocol itself to request that a Job be stapled.  
4898 However, an existing document data formatter might be able to request that the document be  
4899 stapled directly with an embedded instruction within the document data. In this case, the IPP  
4900 implementation does not "support" stapling, however the end user is still able to have some  
4901 control over the stapling of the completed job.
- 4902 3) A Printer object is physically capable of stapling, and an implementation chooses to support  
4903 stapling in the IPP "finishings" attribute. In this case, 'staple' **MUST** be a value in the "finishings-  
4904 supported" Printer object attribute. Doing so, would enable end users to be aware of and make  
4905 use of the stapling feature using IPP attributes.

4906

4907 Even though support for Job Template attributes by a Printer object is **OPTIONAL**, it is  
4908 **RECOMMENDED** that if the device behind a Printer object is capable of realizing any feature or  
4909 function that corresponds to an IPP attribute and some associated value, then that implementation  
4910 **SHOULD** support that IPP attribute and value.

4911 The set of values in any of the supported value attributes is set (populated) by some administrative  
4912 process or automatic sensing mechanism that is outside the scope of this IPP/1.1 document. For  
4913 administrative policy and control reasons, an administrator may choose to make only a subset of possible  
4914 values visible to the end user. In this case, the real output device behind the IPP Printer abstraction may  
4915 be capable of a certain feature, however an administrator is specifying that access to that feature not be  
4916 exposed to the end user through the IPP protocol. Also, since a Printer object may represent a logical  
4917 print device (not just a physical device) the actual process for supporting a value is undefined and left up  
4918 to the implementation. However, if a Printer object supports a value, some manual human action may be  
4919 needed to realize the semantic action associated with the value, but no end user action is required.

4920 For example, if one of the values in the "finishings-supported" attribute is 'staple', the actual process  
4921 might be an automatic staple action by a physical device controlled by some command sent to the  
4922 device. Or, the actual process of stapling might be a manual action by an operator at an operator  
4923 attended Printer object.

4924 For another example of how supported attributes function, consider a system administrator who desires  
4925 to control all print jobs so that no job sheets are printed in order to conserve paper. To force no job  
4926 sheets, the system administrator sets the only supported value for the "job-sheets-supported" attribute to  
4927 'none'. In this case, if a client requests anything except 'none', the create request is rejected or the "job-  
4928 sheets" value is ignored (depending on the value of "ipp-attribute-fidelity"). To force the use of job  
4929 start/end sheets on all jobs, the administrator does not include the value 'none' in the "job-sheets-  
4930 supported" attribute. In this case, if a client requests 'none', the create request is rejected or the "job-  
4931 sheets" value is ignored (again depending on the value of "ipp-attribute-fidelity").

#### 4932 13.2.4 print-stream page

4933 A "print-stream page" is a page according to the definition of pages in the language used to express the  
4934 document data.

#### 4935 13.2.5 impression

4936 An "impression" is the image (possibly many print-stream pages in different configurations) imposed  
4937 onto a single media page.

### 4938 14. APPENDIX B: Status Codes and Suggested Status Code Messages

4939 This section defines status code enum keywords and values that are used to provide semantic  
4940 information on the results of an operation request. Each operation response **MUST** include a status  
4941 code. The response **MAY** also contain a status message that provides a short textual description of the  
4942 status. The status code is intended for use by automata, and the status message is intended for the human  
4943 end user. Since the status message is an **OPTIONAL** component of the operation response, an IPP  
4944 application (i.e., a browser, GUI, print driver or gateway) is **NOT REQUIRED** to examine or display the  
4945 status message, since it **MAY** not be returned to the application.

4946 The prefix of the status keyword defines the class of response as follows:

4947 "informational" - Request received, continuing process  
4948 "successful" - The action was successfully received, understood, and accepted  
4949 "redirection" - Further action must be taken in order to complete the request  
4950 "client-error" - The request contains bad syntax or cannot be fulfilled  
4951 "server-error" - The IPP object failed to fulfill an apparently valid request  
4952

4953 As with type2 enums, IPP status codes are extensible. IPP clients are **NOT REQUIRED** to understand  
4954 the meaning of all registered status codes, though such understanding is obviously desirable. However,

4955 IPP clients MUST understand the class of any status code, as indicated by the prefix, and treat any  
4956 unrecognized response as being equivalent to the first status code of that class, with the exception that an  
4957 unrecognized response MUST NOT be cached. For example, if an unrecognized status code of "client-  
4958 error-xxx-yyy" is received by the client, it can safely assume that there was something wrong with its  
4959 request and treat the response as if it had received a "client-error-bad-request" status code. In such cases,  
4960 IPP applications SHOULD present the OPTIONAL message (if present) to the end user since the  
4961 message is likely to contain human readable information which will help to explain the unusual status.  
4962 The name of the enum is the suggested status message for US English.

4963 The status code values range from 0x0000 to 0x7FFF. The value ranges for each status code class are as  
4964 follows:

4965 "successful" - 0x0000 to 0x00FF  
4966 "informational" - 0x0100 to 0x01FF  
4967 "redirection" - 0x0200 to 0x02FF  
4968 "client-error" - 0x0400 to 0x04FF  
4969 "server-error" - 0x0500 to 0x05FF  
4970

4971 The top half (128 values) of each range (0x0n40 to 0x0nFF, for n = 0 to 5) is reserved for private use  
4972 within each status code class. Values 0x0600 to 0x7FFF are reserved for future assignment and MUST  
4973 NOT be used.

## 4974 14.1 Status Codes

4975 Each status code is described below. Section 14.1.5.9 contains a table that indicates which status codes  
4976 apply to which operations. The Implementer's Guide [IPP-IIG] describe the suggested steps for  
4977 processing IPP attributes for all operations, including returning status codes.

### 4978 14.1.1 Informational

4979 This class of status code indicates a provisional response and is to be used for informational purposes  
4980 only.

4981 There are no status codes defined in IPP/1.1 for this class of status code.

### 4982 14.1.2 Successful Status Codes

4983 This class of status code indicates that the client's request was successfully received, understood, and  
4984 accepted.

#### 4985 14.1.2.1 successful-ok (0x0000)

4986 The request has succeeded and no request attributes were substituted or ignored. In the case of a  
4987 response to a create request, the 'successful-ok' status code indicates that the request was successfully  
4988 received and validated, and that the Job object has been created; it does not indicate that the job has been

4989 processed. The transition of the Job object into the 'completed' state is the only indicator that the job has  
4990 been printed.

#### 4991 14.1.2.2 successful-ok-ignored-or-substituted-attributes (0x0001)

4992 The request has succeeded, but some supplied (1) attributes were ignored or (2) unsupported values were  
4993 substituted with supported values or were ignored in order to perform the operation without rejecting it.  
4994 Unsupported attributes, attribute syntaxes, or values **MUST** be returned in the Unsupported Attributes  
4995 group of the response for all operations. There is an exception to this rule for the query operations: Get-  
4996 Printer-Attributes, Get-Jobs, and Get-Job-Attributes for the "requested-attributes" operation attribute  
4997 only. When the supplied values of the "requested-attributes" operation attribute are requesting attributes  
4998 that are not supported, the IPP object **MAY**, but is **NOT REQUIRED** to, return the "requested-attributes"  
4999 attribute in the Unsupported Attribute response group (with the unsupported values only). See section  
5000 3.2.1.2.

#### 5001 14.1.2.3 successful-ok-conflicting-attributes (0x0002)

5002 The request has succeeded, but some supplied attribute values conflicted with the values of other  
5003 supplied attributes. These conflicting values were either (1) substituted with (supported) values or (2)  
5004 the attributes were removed in order to process the job without rejecting it. Attributes or values which  
5005 conflict with other attributes and have been substituted or ignored **MUST** be returned in the Unsupported  
5006 Attributes group of the response for all operations as supplied by the client. See section 3.2.1.2.

#### 5007 14.1.3 Redirection Status Codes

5008 This class of status code indicates that further action needs to be taken to fulfill the request.

5009 There are no status codes defined in IPP/1.1 for this class of status code.

#### 5010 14.1.4 Client Error Status Codes

5011 This class of status code is intended for cases in which the client seems to have erred. The IPP object  
5012 **SHOULD** return a message containing an explanation of the error situation and whether it is a temporary  
5013 or permanent condition.

#### 5014 14.1.4.1 client-error-bad-request (0x0400)

5015 The request could not be understood by the IPP object due to malformed syntax (such as the value of a  
5016 fixed length attribute whose length does not match the prescribed length for that attribute - see the  
5017 Implementer's Guide [IPP-IIG] ). The IPP application **SHOULD NOT** repeat the request without  
5018 modifications.

## 5019 14.1.4.2 client-error-forbidden (0x0401)

5020 The IPP object understood the request, but is refusing to fulfill it. Additional authentication information  
5021 or authorization credentials will not help and the request SHOULD NOT be repeated. This status code  
5022 is commonly used when the IPP object does not wish to reveal exactly why the request has been refused  
5023 or when no other response is applicable.

## 5024 14.1.4.3 client-error-not-authenticated (0x0402)

5025 The request requires user authentication. The IPP client may repeat the request with suitable  
5026 authentication information. If the request already included authentication information, then this status  
5027 code indicates that authorization has been refused for those credentials. If this response contains the  
5028 same challenge as the prior response, and the user agent has already attempted authentication at least  
5029 once, then the response message may contain relevant diagnostic information. This status codes reveals  
5030 more information than "client-error-forbidden".

## 5031 14.1.4.4 client-error-not-authorized (0x0403)

5032 The requester is not authorized to perform the request. Additional authentication information or  
5033 authorization credentials will not help and the request SHOULD NOT be repeated. This status code is  
5034 used when the IPP object wishes to reveal that the authentication information is understandable,  
5035 however, the requester is explicitly not authorized to perform the request. This status codes reveals  
5036 more information than "client-error-forbidden" and "client-error-not-authenticated".

## 5037 14.1.4.5 client-error-not-possible (0x0404)

5038 This status code is used when the request is for something that can not happen. For example, there  
5039 might be a request to cancel a job that has already been canceled or aborted by the system. The IPP  
5040 client SHOULD NOT repeat the request.

## 5041 14.1.4.6 client-error-timeout (0x0405)

5042 The client did not produce a request within the time that the IPP object was prepared to wait. For  
5043 example, a client issued a Create-Job operation and then, after a long period of time, issued a Send-  
5044 Document operation and this error status code was returned in response to the Send-Document request  
5045 (see section 3.3.1). The IPP object might have been forced to clean up resources that had been held for  
5046 the waiting additional Documents. The IPP object was forced to close the Job since the client took too  
5047 long. The client SHOULD NOT repeat the request without modifications.

## 5048 14.1.4.7 client-error-not-found (0x0406)

5049 The IPP object has not found anything matching the request URI. No indication is given of whether the  
5050 condition is temporary or permanent. For example, a client with an old reference to a Job (a URI) tries  
5051 to cancel the Job, however in the mean time the Job might have been completed and all record of it at the  
5052 Printer has been deleted. This status code, 'client-error-not-found' is returned indicating that the

5053 referenced Job can not be found. This error status code is also used when a client supplies a URI as a  
5054 reference to the document data in either a Print-URI or Send-URI operation, but the document can not be  
5055 found.

5056 In practice, an IPP application should avoid a not found situation by first querying and presenting a list  
5057 of valid Printer URIs and Job URIs to the end-user.

#### 5058 14.1.4.8 client-error-gone (0x0407)

5059 The requested object is no longer available and no forwarding address is known. This condition should  
5060 be considered permanent. Clients with link editing capabilities should delete references to the request  
5061 URI after user approval. If the IPP object does not know or has no facility to determine, whether or not  
5062 the condition is permanent, the status code "client-error-not-found" should be used instead.

5063 This response is primarily intended to assist the task of maintenance by notifying the recipient that the  
5064 resource is intentionally unavailable and that the IPP object administrator desires that remote links to  
5065 that resource be removed. It is not necessary to mark all permanently unavailable resources as "gone" or  
5066 to keep the mark for any length of time -- that is left to the discretion of the IPP object administrator.

#### 5067 14.1.4.9 client-error-request-entity-too-large (0x0408)

5068 The IPP object is refusing to process a request because the request entity is larger than the IPP object is  
5069 willing or able to process. An IPP Printer returns this status code when it limits the size of print jobs and  
5070 it receives a print job that exceeds that limit or when the attributes are so many that their encoding  
5071 causes the request entity to exceed IPP object capacity.

#### 5072 14.1.4.10 client-error-request-value-too-long (0x0409)

5073 The IPP object is refusing to service the request because one or more of the client-supplied attributes has  
5074 a variable length value that is longer than the maximum length specified for that attribute. The IPP  
5075 object might not have sufficient resources (memory, buffers, etc.) to process (even temporarily),  
5076 interpret, and/or ignore a value larger than the maximum length. Another use of this error code is when  
5077 the IPP object supports the processing of a large value that is less than the maximum length, but during  
5078 the processing of the request as a whole, the object may pass the value onto some other system  
5079 component which is not able to accept the large value. For more details, see the Implementer's Guide  
5080 [IPP-IIG] .

5081 Note: For attribute values that are URIs, this rare condition is only likely to occur when a client has  
5082 improperly submitted a request with long query information (e.g. an IPP application allows an end-user  
5083 to enter an invalid URI), when the client has descended into a URI "black hole" of redirection (e.g., a  
5084 redirected URI prefix that points to a suffix of itself), or when the IPP object is under attack by a client  
5085 attempting to exploit security holes present in some IPP objects using fixed-length buffers for reading or  
5086 manipulating the Request-URI.

## 5087 14.1.4.11 client-error-document-format-not-supported (0x040A)

5088 The IPP object is refusing to service the request because the document data is in a format, as specified in  
5089 the "document-format" operation attribute, that is not supported by the Printer object. This error is  
5090 returned independent of the client-supplied "ipp-attribute-fidelity". The Printer object MUST return this  
5091 status code, even if there are other attributes that are not supported as well, since this error is a bigger  
5092 problem than with Job Template attributes.

## 5093 14.1.4.12 client-error-attributes-or-values-not-supported (0x040B)

5094 In a create request, if the Printer object does not support one or more attributes, attribute syntaxes, or  
5095 attribute values supplied in the request and the client supplied the "ipp-attributes-fidelity" operation  
5096 attribute with the 'true' value, the Printer object MUST return this status code. For example, if the  
5097 request indicates 'iso-a4' media, but that media type is not supported by the Printer object. Or, if the  
5098 client supplies an optional attribute and the attribute itself is not even supported by the Printer. If the  
5099 "ipp-attribute-fidelity" attribute is 'false', the Printer MUST ignore or substitute values for unsupported  
5100 attributes and values rather than reject the request and return this status code.

5101 For any operation where a client requests attributes (such as a Get-Jobs, Get-Printer-Attributes, or Get-  
5102 Job-Attributes operation), if the IPP object does not support one or more of the requested attributes, the  
5103 IPP object simply ignores the unsupported requested attributes and processes the request as if they had  
5104 not been supplied, rather than returning this status code. In this case, the IPP object MUST return the  
5105 'successful-ok-ignored-or-substituted-attributes' status code and MAY return the unsupported attributes  
5106 as values of the "requested-attributes" in the Unsupported Attributes Group (see section 14.1.2.2).

## 5107 14.1.4.13 client-error-uri-scheme-not-supported (0x040C)

5108 The type of the client supplied URI in a Print-URI or a Send-URI operation is not supported.

## 5109 14.1.4.14 client-error-charset-not-supported (0x040D)

5110 For any operation, if the IPP Printer does not support the charset supplied by the client in the "attributes-  
5111 charset" operation attribute, the Printer MUST reject the operation and return this status and any 'text' or  
5112 'name' attributes using the 'utf-8' charset (see Section 3.1.4.1).

## 5113 14.1.4.15 client-error-conflicting-attributes (0x040E)

5114 The request is rejected because some attribute values conflicted with the values of other attributes which  
5115 this specification does not permit to be substituted or ignored.

## 5116 14.1.5 Server Error Status Codes

5117 This class of status codes indicates cases in which the IPP object is aware that it has erred or is incapable  
5118 of performing the request. The IPP object SHOULD include a message containing an explanation of the  
5119 error situation, and whether it is a temporary or permanent condition.

## 5120 14.1.5.1 server-error-internal-error (0x0500)

5121 The IPP object encountered an unexpected condition that prevented it from fulfilling the request. This  
5122 error status code differs from "server-error-temporary-error" in that it implies a more permanent type of  
5123 internal error. It also differs from "server-error-device-error" in that it implies an unexpected condition  
5124 (unlike a paper-jam or out-of-toner problem which is undesirable but expected). This error status code  
5125 indicates that probably some knowledgeable human intervention is required.

## 5126 14.1.5.2 server-error-operation-not-supported (0x0501)

5127 The IPP object does not support the functionality required to fulfill the request. This is the appropriate  
5128 response when the IPP object does not recognize an operation or is not capable of supporting it.

## 5129 14.1.5.3 server-error-service-unavailable (0x0502)

5130 The IPP object is currently unable to handle the request due to a temporary overloading or maintenance  
5131 of the IPP object. The implication is that this is a temporary condition which will be alleviated after  
5132 some delay. If known, the length of the delay may be indicated in the message. If no delay is given, the  
5133 IPP application should handle the response as it would for a "server-error-temporary-error" response. If  
5134 the condition is more permanent, the error status codes "client-error-gone" or "client-error-not-found"  
5135 could be used.

## 5136 14.1.5.4 server-error-version-not-supported (0x0503)

5137 The IPP object does not support, or refuses to support, the IPP protocol version that was used in the  
5138 request message. The IPP object is indicating that it is unable or unwilling to complete the request using  
5139 the same version as supplied in the request other than with this error message. The response should  
5140 contain a Message describing why that version is not supported and what other versions are supported by  
5141 that IPP object.

5142 A conforming IPP/1.1 client MUST specify a valid version ('1.1' or '1.0') on each request. A conforming  
5143 IPP/1.1 object MUST NOT return this status code to a conforming IPP/1.1 or IPP/1.0 client. An IPP  
5144 object MUST return this status code to a non-conforming IPP client. The response MUST identify in the  
5145 "version-number" operation attribute the closest version number that the IPP object does support. For  
5146 example, if a client supplies version '1.0', a conforming IPP/1.1 object MUST respond with version '1.0'.

## 5147 14.1.5.5 server-error-device-error (0x0504)

5148 A printer error, such as a paper jam, occurs while the IPP object processes a Print or Send operation.  
5149 The response contains the true Job Status (the values of the "job-state" and "job-state-reasons"  
5150 attributes). Additional information can be returned in the optional "job-state-message" attribute value or  
5151 in the OPTIONAL status message that describes the error in more detail. This error status code is only  
5152 returned in situations where the Printer is unable to accept the create request because of such a device  
5153 error. For example, if the Printer is unable to spool, and can only accept one job at a time, the reason it  
5154 might reject a create request is that the printer currently has a paper jam. In many cases however, where

5155 the Printer object can accept the request even though the Printer has some error condition, the  
5156 'successful-ok' status code will be returned. In such a case, the client would look at the returned Job  
5157 Object Attributes or later query the Printer to determine its state and state reasons.

5158 14.1.5.6 server-error-temporary-error (0x0505)

5159 A temporary error such as a buffer full write error, a memory overflow (i.e. the document data exceeds  
5160 the memory of the Printer), or a disk full condition, occurs while the IPP Printer processes an operation.  
5161 The client MAY try the unmodified request again at some later point in time with an expectation that the  
5162 temporary internal error condition may have been cleared. Alternatively, as an implementation option, a  
5163 Printer object MAY delay the response until the temporary condition is cleared so that no error is  
5164 returned.

5165 14.1.5.7 server-error-not-accepting-jobs (0x0506)

5166 A temporary error indicating that the Printer is not currently accepting jobs, because the administrator  
5167 has set the value of the Printer's "printer-is-not-accepting-jobs" attribute to 'false' (by means outside the  
5168 scope of this IPP/1.1 document).

5169 14.1.5.8 server-error-busy (0x0507)

5170 A temporary error indicating that the Printer is too busy processing jobs and/or other requests. The client  
5171 SHOULD try the unmodified request again at some later point in time with an expectation that the  
5172 temporary busy condition will have been cleared.

5173 14.1.5.9 server-error-job-canceled (0x0508)

5174 An error indicating that the job has been canceled by an operator or the system while the client was  
5175 transmitting the data to the IPP Printer. If a job-id and job-uri had been created, then they are returned in  
5176 the Print-Job, Send-Document, or Send-URI response as usual; otherwise, no job-id and job-uri are  
5177 returned in the response.

## 5178 14.2 Status Codes for IPP Operations

5179 PJ = Print-Job, PU = Print-URI, CJ = Create-Job, SD = Send-Document  
 5180 SU = Send-URI, V = Validate-Job, GA = Get-Job-Attributes and  
 5181 Get-Printer-Attributes, GJ = Get-Jobs, C = Cancel-Job

5182		IPP Operations								
5183	IPP Status Keyword	PJ	PU	CJ	SD	SU	V	GA	GJ	C
5184	-----	--	--	--	--	--	--	--	--	--
5185	successful-ok	x	x	x	x	x	x	x	x	x
5186	successful-ok-ignored-or-substituted-	x	x	x	x	x	x	x	x	x
5187	attributes									
5188	successful-ok-conflicting-attributes	x	x	x	x	x	x	x	x	x
5189	client-error-bad-request	x	x	x	x	x	x	x	x	x
5190	client-error-forbidden	x	x	x	x	x	x	x	x	x
5191	client-error-not-authenticated	x	x	x	x	x	x	x	x	x
5192	client-error-not-authorized	x	x	x	x	x	x	x	x	x
5193	client-error-not-possible	x	x	x	x	x	x	x	x	x
5194	client-error-timeout				x	x				
5195	client-error-not-found	x	x	x	x	x	x	x	x	x
5196	client-error-gone	x	x	x	x	x	x	x	x	x
5197	client-error-request-entity-too-large	x	x	x	x	x	x	x	x	x
5198	client-error-request-value-too-long	x	x	x	x	x	x	x	x	x
5199	client-error-document-format-not-	x	x		x	x	x	x		
5200	supported									
5201	client-error-attributes-or-values-not-	x	x	x	x	x	x	x	x	x
5202	supported									
5203	client-error-uri-scheme-not-supported		x			x				
5204	client-error-charset-not-supported	x	x	x	x	x	x	x	x	x
5205	client-error-conflicting-attributes	x	x	x	x	x	x	x	x	x
5206	server-error-internal-error	x	x	x	x	x	x	x	x	x
5207	server-error-operation-not-supported		x	x	x	x				
5208	server-error-service-unavailable	x	x	x	x	x	x	x	x	x
5209	server-error-version-not-supported	x	x	x	x	x	x	x	x	x
5210	server-error-device-error	x	x	x	x	x				
5211	server-error-temporary-error	x	x	x	x	x				
5212	server-error-not-accepting-jobs	x	x	x			x			
5213	server-error-busy	x	x	x	x	x	x	x	x	x
5214	server-error-job-canceled	x			x					
5215										
5216										

5217 HJ = Hold-Job, RJ = Release-Job, RS = Restart-Job  
 5218 PP = Pause-Printer, RP = Resume-Printer, PJ = Purge-Jobs

5219		IPP Operations (cont.)					
5220	IPP Status Keyword	HJ	RJ	RS	PP	RP	PJ
5221	-----	--	--	--	--	--	--
5222	successful-ok	x	x	x	x	x	x
5223	successful-ok-ignored-or-substituted-	x	x	x	x	x	x
5224	attributes						
5225	successful-ok-conflicting-attributes	x	x	x	x	x	x
5226	client-error-bad-request	x	x	x	x	x	x
5227	client-error-forbidden	x	x	x	x	x	x
5228	client-error-not-authenticated	x	x	x	x	x	x
5229	client-error-not-authorized	x	x	x	x	x	x
5230	client-error-not-possible	x	x	x	x	x	x
5231	client-error-timeout						
5232	client-error-not-found	x	x	x	x	x	x
5233	client-error-gone	x	x	x	x	x	x
5234	client-error-request-entity-too-large	x	x	x	x	x	x
5235	client-error-request-value-too-long	x	x	x	x	x	x
5236	client-error-document-format-not-						
5237	supported						
5238	client-error-attributes-or-values-not-	x	x	x	x	x	x
5239	supported						
5240	client-error-uri-scheme-not-supported						
5241	client-error-charset-not-supported	x	x	x	x	x	x
5242	client-error-conflicting-attributes	x	x	x	x	x	x
5243	server-error-internal-error	x	x	x	x	x	x
5244	server-error-operation-not-supported	x	x	x	x	x	x
5245	server-error-service-unavailable	x	x	x	x	x	x
5246	server-error-version-not-supported	x	x	x	x	x	x
5247	server-error-device-error						
5248	server-error-temporary-error						
5249	server-error-not-accepting-jobs						
5250	server-error-busy	x	x	x	x	x	x
5251	server-error-job-canceled						
5252							

5253

## 5254 15. APPENDIX C: "media" keyword values

5255 Standard keyword values are taken from several sources.

5256 Standard values are defined (taken from DPA[ISO10175] and the Printer MIB[RFC1759]):

- 5257 'default': The default medium for the output device
- 5258 'iso-a4-white': Specifies the ISO A4 white medium
- 5259 'iso-a4-colored': Specifies the ISO A4 colored medium
- 5260 'iso-a4-transparent': Specifies the ISO A4 transparent medium
- 5261 'iso-a3-white': Specifies the ISO A3 white medium
- 5262 'iso-a3-colored': Specifies the ISO A3 colored medium
- 5263 'iso-a5-white': Specifies the ISO A5 white medium
- 5264 'iso-a5-colored': Specifies the ISO A5 colored medium
- 5265 'iso-b4-white': Specifies the ISO B4 white medium
- 5266 'iso-b4-colored': Specifies the ISO B4 colored medium
- 5267 'iso-b5-white': Specifies the ISO B5 white medium
- 5268 'iso-b5-colored': Specifies the ISO B5 colored medium
- 5269 'jis-b4-white': Specifies the JIS B4 white medium
- 5270 'jis-b4-colored': Specifies the JIS B4 colored medium
- 5271 'jis-b5-white': Specifies the JIS B5 white medium
- 5272 'jis-b5-colored': Specifies the JIS B5 colored medium

5273

5274 The following standard values are defined for North American media:

- 5275 'na-letter-white': Specifies the North American letter white medium
- 5276 'na-letter-colored': Specifies the North American letter colored medium
- 5277 'na-letter-transparent': Specifies the North American letter transparent medium
- 5278 'na-legal-white': Specifies the North American legal white medium
- 5279 'na-legal-colored': Specifies the North American legal colored medium

5280

5281 The following standard values are defined for envelopes:

- 5282 'iso-b4-envelope': Specifies the ISO B4 envelope medium
- 5283 'iso-b5-envelope': Specifies the ISO B5 envelope medium
- 5284 'iso-c3-envelope': Specifies the ISO C3 envelope medium
- 5285 'iso-c4-envelope': Specifies the ISO C4 envelope medium
- 5286 'iso-c5-envelope': Specifies the ISO C5 envelope medium
- 5287 'iso-c6-envelope': Specifies the ISO C6 envelope medium
- 5288 'iso-designated-long-envelope': Specifies the ISO Designated Long envelope medium
- 5289 'na-10x13-envelope': Specifies the North American 10x13 envelope medium
- 5290 'na-9x12-envelope': Specifies the North American 9x12 envelope medium

5291 'monarch-envelope': Specifies the Monarch envelope  
5292 'na-number-10-envelope': Specifies the North American number 10 business envelope medium  
5293 'na-7x9-envelope': Specifies the North American 7x9 inch envelope  
5294 'na-9x11-envelope': Specifies the North American 9x11 inch envelope  
5295 'na-10x14-envelope': Specifies the North American 10x14 inch envelope  
5296 'na-number-9-envelope': Specifies the North American number 9 business envelope  
5297 'na-6x9-envelope': Specifies the North American 6x9 inch envelope  
5298 'na-10x15-envelope': Specifies the North American 10x15 inch envelope  
5299

5300 The following standard values are defined for the less commonly used media (white-only):

5301 'executive-white': Specifies the white executive medium  
5302 'folio-white': Specifies the folio white medium  
5303 'invoice-white': Specifies the white invoice medium  
5304 'ledger-white': Specifies the white ledger medium  
5305 'quarto-white': Specifies the white quarto medium  
5306 'iso-a0-white': Specifies the ISO A0 white medium  
5307 'iso-a1-white': Specifies the ISO A1 white medium  
5308 'iso-a2-white': Specifies the ISO A2 white medium  
5309 'iso-a6-white': Specifies the ISO A6 white medium  
5310 'iso-a7-white': Specifies the ISO A7 white medium  
5311 'iso-a8-white': Specifies the ISO A8 white medium  
5312 'iso-a9-white': Specifies the ISO A9 white medium  
5313 'iso-10-white': Specifies the ISO A10 white medium  
5314 'iso-b0-white': Specifies the ISO B0 white medium  
5315 'iso-b1-white': Specifies the ISO B1 white medium  
5316 'iso-b2-white': Specifies the ISO B2 white medium  
5317 'iso-b3-white': Specifies the ISO B3 white medium  
5318 'iso-b6-white': Specifies the ISO B6 white medium  
5319 'iso-b7-white': Specifies the ISO B7 white medium  
5320 'iso-b8-white': Specifies the ISO B8 white medium  
5321 'iso-b9-white': Specifies the ISO B9 white medium  
5322 'iso-b10-white': Specifies the ISO B10 white medium  
5323 'jis-b0-white': Specifies the JIS B0 white medium  
5324 'jis-b1-white': Specifies the JIS B1 white medium  
5325 'jis-b2-white': Specifies the JIS B2 white medium  
5326 'jis-b3-white': Specifies the JIS B3 white medium  
5327 'jis-b6-white': Specifies the JIS B6 white medium  
5328 'jis-b7-white': Specifies the JIS B7 white medium  
5329 'jis-b8-white': Specifies the JIS B8 white medium  
5330 'jis-b9-white': Specifies the JIS B9 white medium  
5331 'jis-b10-white': Specifies the JIS B10 white medium  
5332

5333 The following standard values are defined for engineering media:

5334       `a`: Specifies the engineering A size medium  
5335       `b`: Specifies the engineering B size medium  
5336       `c`: Specifies the engineering C size medium  
5337       `d`: Specifies the engineering D size medium  
5338       `e`: Specifies the engineering E size medium  
5339

5340       The following standard values are defined for input-trays (from ISO DPA and the Printer MIB):

5341       `top`: The top input tray in the printer.  
5342       `middle`: The middle input tray in the printer.  
5343       `bottom`: The bottom input tray in the printer.  
5344       `envelope`: The envelope input tray in the printer.  
5345       `manual`: The manual feed input tray in the printer.  
5346       `large-capacity`: The large capacity input tray in the printer.  
5347       `main`: The main input tray  
5348       `side`: The side input tray  
5349

5350       The following standard values are defined for media sizes (from ISO DPA):

5351       `iso-a0`: Specifies the ISO A0 size: 841 mm by 1189 mm as defined in ISO 216  
5352       `iso-a1`: Specifies the ISO A1 size: 594 mm by 841 mm as defined in ISO 216  
5353       `iso-a2`: Specifies the ISO A2 size: 420 mm by 594 mm as defined in ISO 216  
5354       `iso-a3`: Specifies the ISO A3 size: 297 mm by 420 mm as defined in ISO 216  
5355       `iso-a4`: Specifies the ISO A4 size: 210 mm by 297 mm as defined in ISO 216  
5356       `iso-a5`: Specifies the ISO A5 size: 148 mm by 210 mm as defined in ISO 216  
5357       `iso-a6`: Specifies the ISO A6 size: 105 mm by 148 mm as defined in ISO 216  
5358       `iso-a7`: Specifies the ISO A7 size: 74 mm by 105 mm as defined in ISO 216  
5359       `iso-a8`: Specifies the ISO A8 size: 52 mm by 74 mm as defined in ISO 216  
5360       `iso-a9`: Specifies the ISO A9 size: 37 mm by 52 mm as defined in ISO 216  
5361       `iso-a10`: Specifies the ISO A10 size: 26 mm by 37 mm as defined in ISO 216  
5362       `iso-b0`: Specifies the ISO B0 size: 1000 mm by 1414 mm as defined in ISO 216  
5363       `iso-b1`: Specifies the ISO B1 size: 707 mm by 1000 mm as defined in ISO 216  
5364       `iso-b2`: Specifies the ISO B2 size: 500 mm by 707 mm as defined in ISO 216  
5365       `iso-b3`: Specifies the ISO B3 size: 353 mm by 500 mm as defined in ISO 216  
5366       `iso-b4`: Specifies the ISO B4 size: 250 mm by 353 mm as defined in ISO 216  
5367       `iso-b5`: Specifies the ISO B5 size: 176 mm by 250 mm as defined in ISO 216  
5368       `iso-b6`: Specifies the ISO B6 size: 125 mm by 176 mm as defined in ISO 216  
5369       `iso-b7`: Specifies the ISO B7 size: 88 mm by 125 mm as defined in ISO 216  
5370       `iso-b8`: Specifies the ISO B8 size: 62 mm by 88 mm as defined in ISO 216  
5371       `iso-b9`: Specifies the ISO B9 size: 44 mm by 62 mm as defined in ISO 216  
5372       `iso-b10`: Specifies the ISO B10 size: 31 mm by 44 mm as defined in ISO 216  
5373       `na-letter`: Specifies the North American letter size: 8.5 inches by 11 inches  
5374       `na-legal`: Specifies the North American legal size: 8.5 inches by 14 inches  
5375       `executive`: Specifies the executive size (7.25 X 10.5 in)  
5376       `folio`: Specifies the folio size (8.5 X 13 in)

5377 'invoice': Specifies the invoice size (5.5 X 8.5 in)  
5378 'ledger': Specifies the ledger size (11 X 17 in)  
5379 'quarto': Specifies the quarto size (8.5 X 10.83 in)  
5380 'iso-c3': Specifies the ISO C3 size: 324 mm by 458 mm as defined in ISO 269  
5381 'iso-c4': Specifies the ISO C4 size: 229 mm by 324 mm as defined in ISO 269  
5382 'iso-c5': Specifies the ISO C5 size: 162 mm by 229 mm as defined in ISO 269  
5383 'iso-c6': Specifies the ISO C6 size: 114 mm by 162 mm as defined in ISO 269  
5384 'iso-designated-long': Specifies the ISO Designated Long size: 110 mm by 220 mm as defined in ISO  
5385 269  
5386 'na-10x13-envelope': Specifies the North American 10x13 size: 10 inches by 13 inches  
5387 'na-9x12-envelope': Specifies the North American 9x12 size: 9 inches by 12 inches  
5388 'na-number-10-envelope': Specifies the North American number 10 business envelope size: 4.125  
5389 inches by 9.5 inches  
5390 'na-7x9-envelope': Specifies the North American 7x9 inch envelope size  
5391 'na-9x11-envelope': Specifies the North American 9x11 inch envelope size  
5392 'na-10x14-envelope': Specifies the North American 10x14 inch envelope size  
5393 'na-number-9-envelope': Specifies the North American number 9 business envelope size  
5394 'na-6x9-envelope': Specifies the North American 6x9 envelope size  
5395 'na-10x15-envelope': Specifies the North American 10x15 envelope size  
5396 'monarch-envelope': Specifies the Monarch envelope size (3.87 x 7.5 in)  
5397 'jis-b0': Specifies the JIS B0 size: 1030mm x 1456mm  
5398 'jis-b1': Specifies the JIS B1 size: 728mm x 1030mm  
5399 'jis-b2': Specifies the JIS B2 size: 515mm x 728mm  
5400 'jis-b3': Specifies the JIS B3 size: 364mm x 515mm  
5401 'jis-b4': Specifies the JIS B4 size: 257mm x 364mm  
5402 'jis-b5': Specifies the JIS B5 size: 182mm x 257mm  
5403 'jis-b6': Specifies the JIS B6 size: 128mm x 182mm  
5404 'jis-b7': Specifies the JIS B7 size: 91mm x 128mm  
5405 'jis-b8': Specifies the JIS B8 size: 64mm x 91mm  
5406 'jis-b9': Specifies the JIS B9 size: 45mm x 64mm  
5407 'jis-b10': Specifies the JIS B10 size: 32mm x 45mm

## 5408 16. APPENDIX D: Processing IPP Attributes

5409 When submitting a print job to a Printer object, the IPP model allows a client to supply operation and  
5410 Job Template attributes along with the document data. These Job Template attributes in the create  
5411 request affect the rendering, production and finishing of the documents in the job. Similar types of  
5412 instructions may also be contained in the document to be printed, that is, embedded within the print data  
5413 itself. In addition, the Printer has a set of attributes that describe what rendering and finishing options  
5414 which are supported by that Printer. This model, which allows for flexibility and power, also introduces  
5415 the potential that at job submission time, these client-supplied attributes may conflict with either:

- 5416 - what the implementation is capable of realizing (i.e., what the Printer supports), as well as
- 5417 - the instructions embedded within the print data itself.

5418

5419 The following sections describe how these two types of conflicts are handled in the IPP model.

## 5420 16.1 Fidelity

5421 If there is a conflict between what the client requests and what a Printer object supports, the client may  
5422 request one of two possible conflict handling mechanisms:

- 5423 1) either reject the job since the job can not be processed exactly as specified, or
  - 5424 2) allow the Printer to make any changes necessary to proceed with processing the Job the best it can.
- 5425

5426 In the first case the client is indicating to the Printer object: "Print the job exactly as specified with no  
5427 exceptions, and if that can't be done, don't even bother printing the job at all." In the second case, the  
5428 client is indicating to the Printer object: "It is more important to make sure the job is printed rather than  
5429 be processed exactly as specified; just make sure the job is printed even if client supplied attributes need  
5430 to be changed or ignored."

5431 The IPP model accounts for this situation by introducing an "ipp-attribute-fidelity" attribute.

5432 In a create request, "ipp-attribute-fidelity" is a boolean operation attribute that is **OPTIONALLY**  
5433 supplied by the client. The value 'true' indicates that total fidelity to client supplied Job Template  
5434 attributes and values is required. The client is requesting that the Job be printed exactly as specified, and  
5435 if that is not possible then the job **MUST** be rejected rather than processed incorrectly. The value 'false'  
5436 indicates that a reasonable attempt to print the Job is acceptable. If a Printer does not support some of  
5437 the client supplied Job Template attributes or values, the Printer **MUST** ignore them or substitute any  
5438 supported value for unsupported values, respectively. The Printer may choose to substitute the default  
5439 value associated with that attribute, or use some other supported value that is similar to the unsupported  
5440 requested value. For example, if a client supplies a "media" value of 'na-letter', the Printer may choose  
5441 to substitute 'iso-a4' rather than a default value of 'envelope'. If the client does not supply the "ipp-  
5442 attribute-fidelity" attribute, the Printer assumes a value of 'false'.

5443 Each Printer implementation **MUST** support both types of "fidelity" printing (that is whether the client  
5444 supplies a value of 'true' or 'false'):

- 5445 - If the client supplies 'false' or does not supply the attribute, the Printer object **MUST** always accept  
5446 the request by ignoring unsupported Job Template attributes and by substituting unsupported  
5447 values of supported Job Template attributes with supported values.
  - 5448 - If the client supplies 'true', the Printer object **MUST** reject the request if the client supplies  
5449 unsupported Job Template attributes.
- 5450

5451 Since a client can always query a Printer to find out exactly what is and is not supported, "ipp-attribute-  
5452 fidelity" set to 'false' is useful when:

- 5453 1) The End-User uses a command line interface to request attributes that might not be supported.
- 5454 2) In a GUI context, if the End User expects the job might be moved to another printer and prefers a  
5455 sub-optimal result to nothing at all.
- 5456 3) The End User just wants something reasonable in lieu of nothing at all.

5457

## 5458 16.2 Page Description Language (PDL) Override

5459 If there is a conflict between the value of an IPP Job Template attribute and a corresponding instruction  
5460 in the document data, the value of the IPP attribute SHOULD take precedence over the document  
5461 instruction. Consider the case where a previously formatted file of document data is sent to an IPP  
5462 Printer. In this case, if the client supplies any attributes at job submission time, the client desires that  
5463 those attributes override the embedded instructions. Consider the case were a previously formatted  
5464 document has embedded in it commands to load 'iso-a4' media. However, the document is passed to an  
5465 end user that only has access to a printer with 'na-letter' media loaded. That end user most likely wants  
5466 to submit that document to an IPP Printer with the "media" Job Template attribute set to 'na-letter'. The  
5467 job submission attribute should take precedence over the embedded PDL instruction. However, until  
5468 companies that supply document data interpreters allow a way for external IPP attributes to take  
5469 precedence over embedded job production instructions, a Printer might not be able to support the  
5470 semantics that IPP attributes override the embedded instructions.

5471 The IPP model accounts for this situation by introducing a "pdl-override-supported" attribute that  
5472 describes the Printer objects capabilities to override instructions embedded in the PDL data stream. The  
5473 value of the "pdl-override-supported" attribute is configured by means outside the scope of this IPP/1.1  
5474 document.

5475 This REQUIRED Printer attribute takes on the following values:

- 5476 - 'attempted': This value indicates that the Printer object attempts to make the IPP attribute values  
5477 take precedence over embedded instructions in the document data, however there is no guarantee.
  - 5478 - 'not-attempted': This value indicates that the Printer object makes no attempt to make the IPP  
5479 attribute values take precedence over embedded instructions in the document data.
- 5480

5481 At job processing time, an implementation that supports the value of 'attempted' might do one of several  
5482 different actions:

- 5483 1) Generate an output device specific command sequence to realize the feature represented by the  
5484 IPP attribute value.
  - 5485 2) Parse the document data itself and replace the conflicting embedded instruction with a new  
5486 embedded instruction that matches the intent of the IPP attribute value.
  - 5487 3) Indicate to the Printer that external supplied attributes take precedence over embedded instructions  
5488 and then pass the external IPP attribute values to the document data interpreter.
  - 5489 4) Anything else that allows for the semantics that IPP attributes override embedded document data  
5490 instructions.
- 5491

5492 Since 'attempted' does not offer any type of guarantee, even though a given Printer object might not do a  
5493 very "good" job of attempting to ensure that IPP attributes take a higher precedence over instructions  
5494 embedded in the document data, it would still be a conforming implementation.

5495 At job processing time, an implementation that supports the value of 'not-attempted' might do one of the  
5496 following actions:

- 5497 1) Simply pre-pend the document data with the PDL instruction that corresponds to the client-  
5498 supplied PDL attribute, such that if the document data also has the same PDL instruction, it will  
5499 override what the Printer object pre-pended. In other words, this implementation is using the  
5500 same implementation semantics for the client-supplied IPP attributes as for the Printer object  
5501 defaults.
- 5502 2) Parse the document data and replace the conflicting embedded instruction with a new embedded  
5503 instruction that approximates, but does not match, the semantic intent of the IPP attribute value.  
5504

5505 Note: The "ipp-attribute-fidelity" attribute applies to the Printer's ability to either accept or reject other  
5506 unsupported Job Template attributes. In other words, if "ipp-attribute-fidelity" is set to 'true', a Job is  
5507 accepted if and only if the client supplied Job Template attributes and values are supported by the  
5508 Printer. Whether these attributes actually affect the processing of the Job when the document data  
5509 contains embedded instructions depends on the ability of the Printer to override the instructions  
5510 embedded in the document data with the semantics of the IPP attributes. If the document data attributes  
5511 can be overridden ("pdl-override-supported" set to 'attempted'), the Printer makes an attempt to use the  
5512 IPP attributes when processing the Job. If the document data attributes can not be overridden ("pdl-  
5513 override-supported" set to 'not-attempted'), the Printer makes no attempt to override the embedded  
5514 document data instructions with the IPP attributes when processing the Job, and hence, the IPP attributes  
5515 may fail to affect the Job processing and output when the corresponding instruction is embedded in the  
5516 document data.

### 5517 16.3 Using Job Template Attributes During Document Processing.

5518 The Printer object uses some of the Job object's Job Template attributes during the processing of the  
5519 document data associated with that job. These include, but are not limited to, "orientation-requested",  
5520 "number-up", "sides", "media", and "copies". The processing of each document in a Job Object MUST  
5521 follow the steps below. These steps are intended only to identify when and how attributes are to be used  
5522 in processing document data and any alternative steps that accomplishes the same effect can be used to  
5523 implement this specification.

- 5524 1. Using the client supplied "document-format" attribute or some form of document format detection  
5525 algorithm (if the value of "document-format" is not specific enough), determine whether or not  
5526 the document data has already been formatted for printing. If the document data has been  
5527 formatted, then go to step 2. Otherwise, the document data MUST be formatted. The formatting  
5528 detection algorithm is implementation defined and is not specified by this specification. The  
5529 formatting of the document data uses the "orientation-requested" attribute to determine how the  
5530 formatted print data should be placed on a print-stream page, see section 4.2.10 for the details.  
5531
- 5532 2. The document data is in the form of a print-stream in a known media type. The "page-ranges"  
5533 attribute is used to select, as specified in section 4.2.7, a sub-sequence of the pages in the print-  
5534 stream that are to be processed and images.  
5535

5536 3. The input to this step is a sequence of print-stream pages. This step is controlled by the "number-  
5537 up" attribute. If the value of "number-up" is N, then during the processing of the print-stream  
5538 pages, each N print-stream pages are positioned, as specified in section 4.2.9, to create a single  
5539 impression. If a given document does not have N more print-stream pages, then the completion  
5540 of the impression is controlled by the "multiple-document-handling" attribute as described in  
5541 section 4.2.4; when the value of this attribute is 'single-document' or 'single-document-new-  
5542 sheet', the print-stream pages of document data from subsequent documents is used to complete  
5543 the impression.

5544  
5545 The size(scaling), position(translation) and rotation of the print-stream pages on the impression is  
5546 implementation defined. Note that during this process the print-stream pages may be rendered to  
5547 a form suitable for placing on the impression; this rendering is controlled by the values of the  
5548 "printer-resolution" and "print-quality" attributes as described in sections 4.2.12 and 4.2.13. In  
5549 the case N=1, the impression is nearly the same as the print-stream page; the differences would  
5550 only be in the size, position and rotation of the print-stream page and/or any decoration, such as a  
5551 frame to the page, that is added by the implementation.

5552  
5553 4. The collection of impressions is placed, in sequence, onto sides of the media sheets. This  
5554 placement is controlled by the "sides" attribute and the orientation of the print-stream page, as  
5555 described in section 4.2.8. The orientation of the print-stream pages affects the orientation of the  
5556 impression; for example, if "number-up" equals 2, then, typically, two portrait print-stream pages  
5557 become one landscape impression. Note that the placement of impressions onto media sheets is  
5558 also controlled by the "multiple-document-handling" attribute as described in section 4.2.4.

5559  
5560 5. The "copies" and "multiple-document-handling" attributes are used to determine how many copies  
5561 of each media instance are created and in what order. See sections 4.2.5 and 4.2.4 for the details.

5562  
5563 6. When the correct number of copies are created, the media instances are finished according to the  
5564 values of the "finishings" attribute as described in 4.2.6. Note that sometimes finishing  
5565 operations may require manual intervention to perform the finishing operations on the copies,  
5566 especially uncollated copies. This specification allows any or all of the processing steps to be  
5567 performed automatically or manually at the discretion of the Printer object.

## 5568 17. APPENDIX E: Generic Directory Schema

5569 This section defines a generic schema for an entry in a directory service. A directory service is a means  
5570 by which service users can locate service providers. In IPP environments, this means that IPP Printers  
5571 can be registered (either automatically or with the help of an administrator) as entries of type printer in  
5572 the directory using an implementation specific mechanism such as entry attributes, entry type fields,  
5573 specific branches, etc. IPP clients can search or browse for entries of type printer. Clients use the  
5574 directory service to find entries based on naming, organizational contexts, or filtered searches on  
5575 attribute values of entries. For example, a client can find all printers in the "Local Department" context.  
5576 Authentication and authorization are also often part of a directory service so that an administrator can

5577 place limits on end users so that they are only allowed to find entries to which they have certain access  
5578 rights. IPP itself does not require any specific directory service protocol or provider.

5579 Note: Some directory implementations allow for the notion of "aliasing". That is, one directory entry  
5580 object can appear as multiple directory entry object with different names for each object. In each case,  
5581 each alias refers to the same directory entry object which refers to a single IPP Printer object.

5582 The generic schema is a subset of IPP Printer Job Template and Printer Description attributes (sections  
5583 4.2 and 4.4). These attributes are identified as either RECOMMENDED or OPTIONAL for the  
5584 directory entry itself. This conformance labeling is NOT the same conformance labeling applied to the  
5585 attributes of IPP Printers objects. The conformance labeling in this Appendix is intended to apply to  
5586 directory templates and to IPP Printer implementations that subscribe by adding one or more entries to a  
5587 directory. RECOMMENDED attributes SHOULD be associated with each directory entry. OPTIONAL  
5588 attributes MAY be associated with the directory entry (if known or supported). In addition, all directory  
5589 entry attributes SHOULD reflect the current attribute values for the corresponding Printer object.

5590 The names of attributes in directory schema and entries SHOULD be the same as the IPP Printer  
5591 attribute names as shown.

5592 In order to bridge between the directory service and the IPP Printer object, one of the RECOMMENDED  
5593 directory entry attributes is the Printer object's "printer-uri-supported" attribute. The IPP client queries  
5594 the "printer-uri-supported" attribute in the directory entry and then addresses the IPP Printer object using  
5595 one of its URIs. The "uri-security-supported" attribute identifies the protocol (if any) used to secure a  
5596 channel.

5597 The following attributes define the generic schema for directory entries of type PRINTER:

5598	printer-uri-supported	RECOMMENDED	Section 4.4.1
5599	uri-security-supported	RECOMMENDED	Section 4.4.2
5600	printer-name	RECOMMENDED	Section 4.4.3
5601	printer-location	RECOMMENDED	Section 4.4.4
5602	printer-info	OPTIONAL	Section 4.4.5
5603	printer-more-info	OPTIONAL	Section 4.4.6
5604	printer-make-and-model	RECOMMENDED	Section 4.4.8
5605	charset-supported	OPTIONAL	Section 4.4.15
5606	generated-natural-language-		
5607	supported	OPTIONAL	Section 4.4.17
5608	document-format-supported	RECOMMENDED	Section 4.4.19
5609	color-supported	RECOMMENDED	Section 4.4.23
5610	finishings-supported	OPTIONAL	Section 4.2.6
5611	number-up-supported	OPTIONAL	Section 4.2.7
5612	sides-supported	RECOMMENDED	Section 4.2.8
5613	media-supported	RECOMMENDED	Section 4.2.11
5614	printer-resolution-supported	OPTIONAL	Section 4.2.12
5615	print-quality-supported	OPTIONAL	Section 4.2.13
5616	pages-per-minute	OPTIONAL	Section 4.4.33
5617	pages-per-minute-color	OPTIONAL	Section 4.4.34

5618

## 5619 18. APPENDIX F: Differences between the IPP/1.0 and IPP/1.1 "Model and Semantics" Specifications

5620 The following IPP/1.0 [IPP-MOD1.0] extensions and clarifications have been incorporated into IPP/1.1:

- 5621 1. Section 3.1.7 - clarified that only the version number parameter will be carried forward into  
5622 future major or minor versions of the protocol.
- 5623 2. Section 3.2.1.1 - clarified that the Printer object rejects a Print-Job request if it does not support  
5624 the "compression" operation attribute and a client supplies it.
- 5625 3. Sections 3.2.7, 3.2.8, and 3.2.9 - added the OPTIONAL Pause-Printer, Resume-Printer, and  
5626 Purge-Jobs operations
- 5627 4. Sections 3.3.5, 3.3.6, and 3.3.7 - added the OPTIONAL Hold-Job, Release-Job, and Restart-Job  
5628 operations.
- 5629 5. Section 4.1.9 - added 'image-tiff' and 'application/pdf' values.
- 5630 6. Section 4.2.2 - added the 'indefinite' keyword value to the "job-hold-until" attribute for use with  
5631 the create operations and Hold-Job and Restart-Job operations.
- 5632 7. Section 4.2.6 - added more enum values to the "finishings" Job Template attribute.
- 5633 8. Section 4.3.7.1 - added the Partitioning of Job States section.
- 5634 9. Section 4.3.8 - added the 'job-restartable' keyword value to the "job-state-reasons" attribute for  
5635 use with the Restart-Job operation.
- 5636 10. Section 4.4.2 - added the 'tls' keyword value to the "uri-security-supported" attribute.
- 5637 11. Section 4.4.11 - added the 'moving-to-paused' keyword value to the "printer-state-reasons"  
5638 attribute for use with the Pause-Job operation.
- 5639 12. Section 4.4.11 - replaced the duplicate 'marker-supply-low' keyword with the missing 'toner-  
5640 empty' keyword for the "printer-state-reasons" attribute.
- 5641 13. Section 4.4.13 - added the enum values to the "operations-supported" attribute for the new  
5642 operations. Clarified that the values of this attribute are encoded as any enum, namely 32-bit  
5643 values.
- 5644 14. Sections 4.4.33 and 4.4.34 - added the OPTIONAL "pages-per-minute" and "pages-per-minute-  
5645 color" Printer Description attributes.
- 5646 15. Section 8.5 - added the security discussion around the new operator operations.
- 5647 16. Section 17 - added the OPTIONAL "pages-per-minute" and "pages-per-minute-color" Printer  
5648 attributes to the Directory schema.

5649 The following changes were made to IPP/1.0 [IPP-MOD1.0] to create this IPP/1.1 document:

- 5650 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  
5651 MUST support version 1.1 and MAY support version 1.0.
- 5652 2. Section 4.1.9 - deleted 'text/plain; charset=iso-10646-ucs-2', since binary is not legal with the  
5653 'text' type.
- 5654 3. Section 5.4, 8.2, and 8.7 - changed the IPP object security requirements from OPTIONAL non-  
5655 standards track SSL3 to RECOMMENDED standards track TLS. Changed the client security

5656 requirements from RECOMMENDED non-standards track SSL3 to RECOMMENDED  
5657 standards track TLS

5658 See also the "IPP/1.1 Encoding and Transport" [ipp-pro] document for differences between IPP/1.0 [IPP-  
5659 PRO1.0] and IPP/1.1 [IPP-PRO].

5660