

1 INTERNET-DRAFT
2 draft-ietf-ipp-model-10.txt

R. deBry
IBM Corporation
T. Hastings
Xerox Corporation
R. Herriot
Sun Microsystems
S. Isaacson
Novell, Inc.
P. Powell
San Diego State University
June 19, 1998

3
4
5
6
7
8
9
10
11
12
13
14
15 Internet Printing Protocol/1.0: Model and Semantics
16 Copyright (C) The Internet Society (date). All Rights Reserved.

17 Status of this Memo

18 This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering
19 Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute
20 working documents as Internet-Drafts.

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

24 To learn the current status of any Internet-Draft, please check the "1id-abstracts.txt" listing contained in
25 the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au
26 (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

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. The protocol is heavily influenced by the printing model
31 introduced in the Document Printing Application (DPA) [ISO10175] standard. Although DPA specifies
32 both end user and administrative features, IPP version 1.0 (IPP/1.0) focuses only on end user
33 functionality.

34 The full set of IPP documents includes:

- 35 Requirements for an Internet Printing Protocol [IPP-REQ]
- 36 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [IPP-RAT]
- 37 Internet Printing Protocol/1.0: Model and Semantics (this document)
- 38 Internet Printing Protocol/1.0: Transport and Encoding [IPP-PRO]

39

40 The requirements document, "Requirements for an Internet Printing Protocol", takes a broad look at
41 distributed printing functionality, and it enumerates real-life scenarios that help to clarify the features that
42 need to be included in a printing protocol for the Internet. It identifies requirements for three types of
43 users: end users, operators, and administrators. The requirements document calls out a subset of end
44 user requirements that are satisfied in IPP/1.0. Operator and administrator requirements are out of scope
45 for version 1.0. The rationale document, "Rationale for the Structure and Model and Protocol for the
46 Internet Printing Protocol", describes IPP from a high level view, defines a roadmap for the various
47 documents that form the suite of IPP specifications, and gives background and rationale for the IETF
48 working group's major decisions. This document, "Internet Printing Protocol/1.0: Model and Semantics",
49 describes a simplified model with abstract objects, their attributes, and their operations. The model
50 introduces a Printer and a Job. The Job supports multiple documents per Job. The model document also
51 addresses how security, internationalization, and directory issues are addressed. The protocol
52 specification, " Internet Printing Protocol/1.0: Transport and Encoding", is a formal mapping of the
53 abstract operations and attributes defined in the model document onto HTTP/1.1. The protocol
54 specification defines the encoding rules for a new Internet media type called "application/ipp".

55 Table of Contents

56	1.	Introduction.....	9
57	1.1	Simplified Printing Model.....	11
58	2.	IPP Objects	13
59	2.1	Printer Object.....	13
60	2.2	Job Object.....	15
61	2.3	Object Relationships.....	16
62	2.4	Object Identity	17
63	3.	IPP Operations	19
64	3.1	Common Semantics	20
65	3.1.1	Required Elements.....	20
66	3.1.2	Operation IDs and Request IDs	21
67	3.1.3	Attributes	21
68	3.1.4	Character Set and Natural Language Operation Attributes	23
69	3.1.4.1	Request Operation Attributes.....	23
70	3.1.4.2	Response Operation Attributes	26
71	3.1.5	Operation Targets.....	27
72	3.1.6	Operation Status Codes and Messages.....	29
73	3.1.7	Versions	29
74	3.1.8	Job Creation Operations	31
75	3.2	Printer Operations.....	33
76	3.2.1	Print-Job Operation	33
77	3.2.1.1	Print-Job Request	33
78	3.2.1.2	Print-Job Response	37
79	3.2.2	Print-URI Operation.....	39
80	3.2.3	Validate-Job Operation.....	40
81	3.2.4	Create-Job Operation	40
82	3.2.5	Get-Printer-Attributes Operation	41
83	3.2.5.1	Get-Printer-Attributes Request	41
84	3.2.5.2	Get-Printer-Attributes Response	43
85	3.2.6	Get-Jobs Operation	44
86	3.2.6.1	Get-Jobs Request.....	44
87	3.2.6.2	Get-Jobs Response	46
88	3.3	Job Operations.....	47
89	3.3.1	Send-Document Operation	47
90	3.3.1.1	Send-Document Request	48
91	3.3.1.2	Send-Document Response	50
92	3.3.2	Send-URI Operation.....	50
93	3.3.3	Cancel-Job Operation	51

94	3.3.3.1	Cancel-Job Request	51
95	3.3.3.2	Cancel-Job Response	51
96	3.3.4	Get-Job-Attributes Operation	52
97	3.3.4.1	Get-Job-Attributes Request	53
98	3.3.4.2	Get-Job-Attributes Response	53
99	4.	Object Attributes	54
100	4.1	Attribute Syntaxes.....	55
101	4.1.1	'text'	56
102	4.1.1.1	'textWithoutLanguage'	56
103	4.1.1.2	'textWithLanguage'	57
104	4.1.2	'name'	57
105	4.1.2.1	'nameWithoutLanguage'	58
106	4.1.2.2	'nameWithLanguage'	58
107	4.1.3	'keyword'	59
108	4.1.4	'enum'	60
109	4.1.5	'uri'	60
110	4.1.6	'uriScheme'	60
111	4.1.7	'charset'	61
112	4.1.8	'naturalLanguage'	61
113	4.1.9	'mimeMediaType'	62
114	4.1.10	'octetString'	63
115	4.1.11	'boolean'	63
116	4.1.12	'integer'	63
117	4.1.13	'rangeOfInteger'	63
118	4.1.14	'dateTime'	64
119	4.1.15	'resolution'	64
120	4.1.16	'1setOf X'	64
121	4.2	Job Template Attributes	64
122	4.2.1	job-priority (integer(1:100)).....	68
123	4.2.2	job-hold-until (type3 keyword name (MAX)).....	69
124	4.2.3	job-sheets (type3 keyword name(MAX))	70
125	4.2.4	multiple-document-handling (type2 keyword)	70
126	4.2.5	copies (integer(1:MAX))	71
127	4.2.6	finishings (1setOf type2 enum).....	71
128	4.2.7	page-ranges (1setOf rangeOfInteger (1:MAX))	72
129	4.2.8	sides (type2 keyword).....	73
130	4.2.9	number-up (integer(1:MAX))	74
131	4.2.10	orientation-requested (type2 enum)	74
132	4.2.11	media (type3 keyword name(MAX)).....	75
133	4.2.12	printer-resolution (resolution).....	76

134	4.2.13	print-quality (type2 enum)	76
135	4.3	Job Description Attributes.....	76
136	4.3.1	job-uri (uri)	78
137	4.3.2	job-id (integer(1:MAX)).....	78
138	4.3.3	job-printer-uri (uri)	78
139	4.3.4	job-more-info (uri).....	79
140	4.3.5	job-name (name(MAX))	79
141	4.3.6	job-originating-user-name (name(MAX)).....	79
142	4.3.7	job-state (type1 enum).....	79
143	4.3.8	job-state-reasons (1setOf type2 keyword).....	82
144	4.3.9	job-state-message (text(MAX))	84
145	4.3.10	number-of-documents (integer(0:MAX))	85
146	4.3.11	output-device-assigned (name(127))	85
147	4.3.12	time-at-creation (integer(0:MAX))	85
148	4.3.13	time-at-processing (integer(0:MAX)).....	85
149	4.3.14	time-at-completed (integer(0:MAX)).....	85
150	4.3.15	number-of-intervening-jobs (integer(0:MAX))	85
151	4.3.16	job-message-from-operator (text(127)).....	86
152	4.3.17	job-k-octets (integer(0:MAX))	86
153	4.3.18	job-impressions (integer(0:MAX)).....	86
154	4.3.19	job-media-sheets (integer(0:MAX))	87
155	4.3.20	job-k-octets-processed (integer(0:MAX)).....	87
156	4.3.21	job-impressions-completed (integer(0:MAX)).....	87
157	4.3.22	job-media-sheets-completed (integer(0:MAX)).....	88
158	4.3.23	attributes-charset (charset).....	88
159	4.3.24	attributes-natural-language (naturalLanguage)	88
160	4.4	Printer Description Attributes.....	88
161	4.4.1	printer-uri-supported (1setOf uri)	90
162	4.4.2	uri-security-supported (1setOf type2 keyword).....	91
163	4.4.3	printer-name (name(127)).....	92
164	4.4.4	printer-location (text(127)).....	92
165	4.4.5	printer-info (text(127))	92
166	4.4.6	printer-more-info (uri)	92
167	4.4.7	printer-driver-installer (uri)	92
168	4.4.8	printer-make-and-model (text(127)).....	93
169	4.4.9	printer-more-info-manufacturer (uri).....	93
170	4.4.10	printer-state (type1 enum)	93
171	4.4.11	printer-state-reasons (1setOf type2 keyword)	94
172	4.4.12	printer-state-message (text(MAX)).....	96
173	4.4.13	operations-supported (1setOf type2 enum)	97

174	4.4.14	charset-configured (charset)	97
175	4.4.15	charset-supported (1setOf charset)	98
176	4.4.16	natural-language-configured (naturalLanguage)	98
177	4.4.17	generated-natural-language-supported (1setOf naturalLanguage).....	98
178	4.4.18	document-format-default (mimeMediaType).....	99
179	4.4.19	document-format-supported (1setOf mimeMediaType).....	99
180	4.4.20	printer-is-accepting-jobs (boolean).....	99
181	4.4.21	queued-job-count (integer(0:MAX)).....	99
182	4.4.22	printer-message-from-operator (text(127))	99
183	4.4.23	color-supported (boolean)	100
184	4.4.24	reference-uri-schemes-supported (1setOf uriScheme)	100
185	4.4.25	pdl-override-supported (type2 keyword).....	100
186	4.4.26	printer-up-time (integer(1:MAX)).....	100
187	4.4.27	printer-current-time (dateTime)	101
188	4.4.28	multiple-operation-time-out (integer(1:MAX))	101
189	4.4.29	compression-supported (1setOf type3 keyword)	101
190	4.4.30	job-k-octets-supported (rangeOfInteger(0:MAX))	102
191	4.4.31	job-impressions-supported (rangeOfInteger(0:MAX)).....	102
192	4.4.32	job-media-sheets-supported (rangeOfInteger(0:MAX)).....	102
193	5.	Conformance	102
194	5.1	Client Conformance Requirements	102
195	5.2	IPP Object Conformance Requirements.....	103
196	5.2.1	Objects	103
197	5.2.2	Operations.....	103
198	5.2.3	IPP Object Attributes	104
199	5.2.4	Extensions.....	104
200	5.2.5	Attribute Syntaxes	104
201	5.3	Charset and Natural Language Requirements	105
202	5.4	Security Conformance Requirements.....	105
203	6.	IANA Considerations (registered and private extensions).....	105
204	6.1	Typed 'keyword' and 'enum' Extensions	106
205	6.2	Attribute Extensibility	108
206	6.3	Attribute Syntax Extensibility	108
207	6.4	Operation Extensibility	109
208	6.5	Status Code Extensibility	109
209	6.6	Registration of MIME types/sub-types for document-formats.....	110
210	6.7	Registration of charsets for use in 'charset' attribute values	110
211	7.	Internationalization Considerations	110
212	8.	Security Considerations	113
213	8.1	Security Scenarios.....	114

214	8.1.1	Client and Server in the Same Security Domain	114
215	8.1.2	Client and Server in Different Security Domains	114
216	8.1.3	Print by Reference	114
217	8.2	URIs for TLS and non-TLS Access.....	115
218	8.3	The "requesting-user-name" (name(MAX)) Operation Attribute.....	115
219	8.4	Restricted Queries.....	117
220	8.5	IPP Security Application Profile for TLS.....	117
221	9.	References.....	117
222	10.	Copyright Notice	120
223	11.	Author's Address	121
224	12.	Formats for IPP Registration Proposals	125
225	12.1	Type2 keyword attribute values registration	125
226	12.2	Type3 keyword attribute values registration	125
227	12.3	Type2 enum attribute values registration	126
228	12.4	Type3 enum attribute values registration	126
229	12.5	Attribute registration.....	126
230	12.6	Attribute Syntax registration	127
231	12.7	Operation registration	127
232	12.8	Status code registration.....	127
233	13.	APPENDIX A: Terminology	128
234	13.1	Conformance Terminology.....	128
235	13.1.1	MUST.....	128
236	13.1.2	MUST NOT.....	128
237	13.1.3	SHOULD.....	128
238	13.1.4	SHOULD NOT.....	129
239	13.1.5	MAY.....	129
240	13.1.6	NEED NOT	129
241	13.2	Model Terminology	129
242	13.2.1	Keyword	129
243	13.2.2	Attributes	129
244	13.2.2.1	Attribute Name.....	130
245	13.2.2.2	Attribute Group Name.....	130
246	13.2.2.3	Attribute Value.....	130
247	13.2.2.4	Attribute Syntax	130
248	13.2.3	Supports.....	130
249	13.2.4	print-stream page.....	132
250	13.2.5	impression	132
251	14.	APPENDIX B: Status Codes and Suggested Status Code Messages	132
252	14.1	Status Codes.....	133
253	14.1.1	Informational.....	133

254	14.1.2	Successful Status Codes	134
255	14.1.2.1	successful-ok (0x0000)	134
256	14.1.2.2	successful-ok-ignored-or-substituted-attributes (0x0001)	134
257	14.1.2.3	successful-ok-conflicting-attributes (0x0002)	134
258	14.1.3	Redirection Status Codes	134
259	14.1.4	Client Error Status Codes	134
260	14.1.4.1	client-error-bad-request (0x0400)	135
261	14.1.4.2	client-error-forbidden (0x0401)	135
262	14.1.4.3	client-error-not-authenticated (0x0402)	135
263	14.1.4.4	client-error-not-authorized (0x0403)	135
264	14.1.4.5	client-error-not-possible (0x0404)	135
265	14.1.4.6	client-error-timeout (0x0405)	135
266	14.1.4.7	client-error-not-found (0x0406)	136
267	14.1.4.8	client-error-gone (0x0407)	136
268	14.1.4.9	client-error-request-entity-too-large (0x0408)	136
269	14.1.4.10	client-error-request-value-too-long (0x0409)	137
270	14.1.4.11	client-error-document-format-not-supported (0x040A)	137
271	14.1.4.12	client-error-attributes-or-values-not-supported (0x040B)	137
272	14.1.4.13	client-error-uri-scheme-not-supported (0x040C)	138
273	14.1.4.14	client-error-charset-not-supported (0x040D)	138
274	14.1.4.15	client-error-conflicting-attributes (0x040E)	138
275	14.1.5	Server Error Status Codes	138
276	14.1.5.1	server-error-internal-error (0x0500)	138
277	14.1.5.2	server-error-operation-not-supported (0x0501)	138
278	14.1.5.3	server-error-service-unavailable (0x0502)	138
279	14.1.5.4	server-error-version-not-supported (0x0503)	139
280	14.1.5.5	server-error-device-error (0x0504)	139
281	14.1.5.6	server-error-temporary-error (0x0505)	139
282	14.1.5.7	server-error-not-accepting-jobs (0x0506)	140
283	14.1.5.8	server-error-busy (0x0507)	140
284	14.2	Status Codes for IPP Operations	141
285	15.	APPENDIX C: "media" keyword values	141
286	16.	APPENDIX D: Processing IPP Attributes	146
287	16.1	Fidelity	146
288	16.2	Page Description Language (PDL) Override	147
289	16.3	Suggested Operation Processing Steps for All Operations	149
290	16.3.1	Validate version number	149
291	16.3.2	Validate operation identifier	150
292	16.3.3	Validate the request identifier	150
293	16.3.4	Validate attribute group and attribute presence and order	150

294	16.3.4.1	Validate the presence and order of attribute groups.....	150
295	16.3.4.2	Ignore unknown attribute groups in the expected position.....	150
296	16.3.4.3	Validate the presence of a single occurrence of required Operation attributes	151
297	16.3.5	Validate the values of the MANDATORY Operation attributes	157
298	16.3.6	Validate the values of the OPTIONAL Operation attributes.....	160
299	16.4	Suggested Additional Processing Steps for Operations that Create/Validate Jobs and	
300		Add Documents	162
301	16.4.1	Default "ipp-attribute-fidelity" if not supplied.....	162
302	16.4.2	Check that the Printer object is accepting jobs	162
303	16.4.3	Validate the values of the Job Template attributes.....	162
304	16.4.4	Check for conflicting Job Template attributes values.....	166
305	16.4.5	Decide whether to REJECT the request.....	167
306	16.4.6	For the Validate-Job operation, RETURN one of the success status codes.....	167
307	16.4.7	Create the Job object with attributes to support	167
308	16.4.8	Return one of the success status codes.....	169
309	16.4.9	Accept appended Document Content.....	169
310	16.4.10	Scheduling and Starting to Process the Job	169
311	16.4.11	Completing the Job.....	170
312	16.4.12	Destroying the Job after completion.....	170
313	16.4.13	Interaction with "ipp-attribute-fidelity"	170
314	16.5	Using Job Template Attributes During Document Processing.	170
315	17.	APPENDIX E: Generic Directory Schema.....	172

316 1. Introduction

317 The Internet Printing Protocol (IPP) is an application level protocol that can be used for distributed
 318 printing using Internet tools and technologies. IPP version 1.0 (IPP/1.0) focuses only on end user
 319 functionality. This document is just one of a suite of documents that fully define IPP. The full set of IPP
 320 documents includes:

- 321 Requirements for an Internet Printing Protocol [IPP-REQ]
- 322 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [IPP-RAT]
- 323 Internet Printing Protocol/1.0: Model and Semantics (this document)
- 324 Internet Printing Protocol/1.0: Transport and Encoding [IPP-PRO]

325
 326 Anyone reading this document for the first time is strongly encouraged to read the IPP documents in the
 327 following order:

- 328 1. The requirements document, "Requirements for an Internet Printing Protocol". That document
 329 takes a broad look at distributed printing functionality, and it enumerates real-life scenarios that

- 330 help to clarify the features that need to be included in a printing protocol for the Internet. It
331 identifies requirements for three types of users: end users, operators, and administrators. The
332 requirements document calls out a subset of end user requirements that are satisfied in IPP/1.0.
333 Operator and administrator requirements are out of scope for version 1.0.
- 334 2. The rationale document, "Rationale for the Structure and Model and Protocol for the Internet
335 Printing Protocol". That document describes IPP from a high level view, defines a roadmap for
336 the various documents that form the suite of IPP specifications, and gives background and
337 rationale for the IETF working group's major decisions.
 - 338 3. This document, the "Internet Printing Protocol/1.0: Model and Semantics" document. This
339 document describes a simplified model with abstract objects, their attributes, and their operations.
340 The model introduces a Printer and a Job. A Job optionally supports multiple documents per Job.
341 The model document also describes how security, internationalization, and directory issues are
342 addressed.
 - 343 4. The protocol specification, " Internet Printing Protocol/1.0: Transport and Encoding". That
344 document is a formal mapping of the abstract operations and attributes defined in the model
345 document onto HTTP/1.1. The protocol specification defines the encoding rules for a new
346 Internet media type called "application/ipp".

347

348 This document is laid out as follows:

- 349 - The rest of Section 1 is an introduction to the IPP simplified model for distributed printing.
- 350 - Section 2 introduces the object types covered in the model with their basic behaviors, attributes, and
351 interactions.
- 352 - Section 3 defines the operations included in IPP/1.0. IPP operations are synchronous, therefore, for
353 each operation, there is a both request and a response.
- 354 - Section 4 defines the attributes (and their syntaxes) that are used in the model.
- 355 - Sections 5 - 6 summarizes the implementation conformance requirements for objects that support
356 the protocol and IANA considerations, respectively.
- 357 - Sections 7 - 11 cover the Internationalization and Security considerations as well as References,
358 Copyright Notice, and Author contact information.
- 359 - Sections 12 - 14 are appendices that cover Terminology, Status Codes and Messages, and "media"
360 keyword values. This document uses terms such as "attributes", "keywords", and "support".
361 These terms have special meaning and are defined in the model terminology section. Capitalized
362 terms such as MANDATORY, SHALL, and OPTIONAL have special meaning relating to
363 conformance. These terms are defined in the section on conformance terminology, most of which
364 is taken from RFC 2119 [RFC2119].
- 365 - Section 15 is an appendix that defines the rules and suggested techniques for the processing of
366 attributes in client requests by IPP objects. This section helps to clarify the effects of interactions
367 between related attributes and their values.

368 - Section 16 is an appendix that enumerates the subset of Printer attributes that form a generic
369 directory schema. These attributes are useful when registering a Printer so that a client can find
370 the Printer not just by name, but by filtered searches as well.

371 1.1 Simplified Printing Model

372 In order to achieve its goal of realizing a workable printing protocol for the Internet, the Internet Printing
373 Protocol (IPP) is based on a simplified printing model that abstracts the many components of real world
374 printing solutions. The Internet is a distributed computing environment where requesters of print services
375 (clients, applications, printer drivers, etc.) cooperate and interact with print service providers. This model
376 and semantics document describes a simple, abstract model for IPP even though the underlying
377 configurations may be complex "n-tier" client/server systems. An important simplifying step in the IPP
378 model is to expose only the key objects and interfaces required for printing. The model described in this
379 model document does not include features, interfaces, and relationships that are beyond the scope of the
380 first version of IPP (IPP/1.0). IPP/1.0 incorporates many of the relevant ideas and lessons learned from
381 other specification and development efforts [HTPP] [ISO10175] [LDPA] [P1387.4] [PSIS] [RFC1179]
382 [SWP].

383 The IPP/1.0 model encapsulates the important components of distributed printing into two object types:

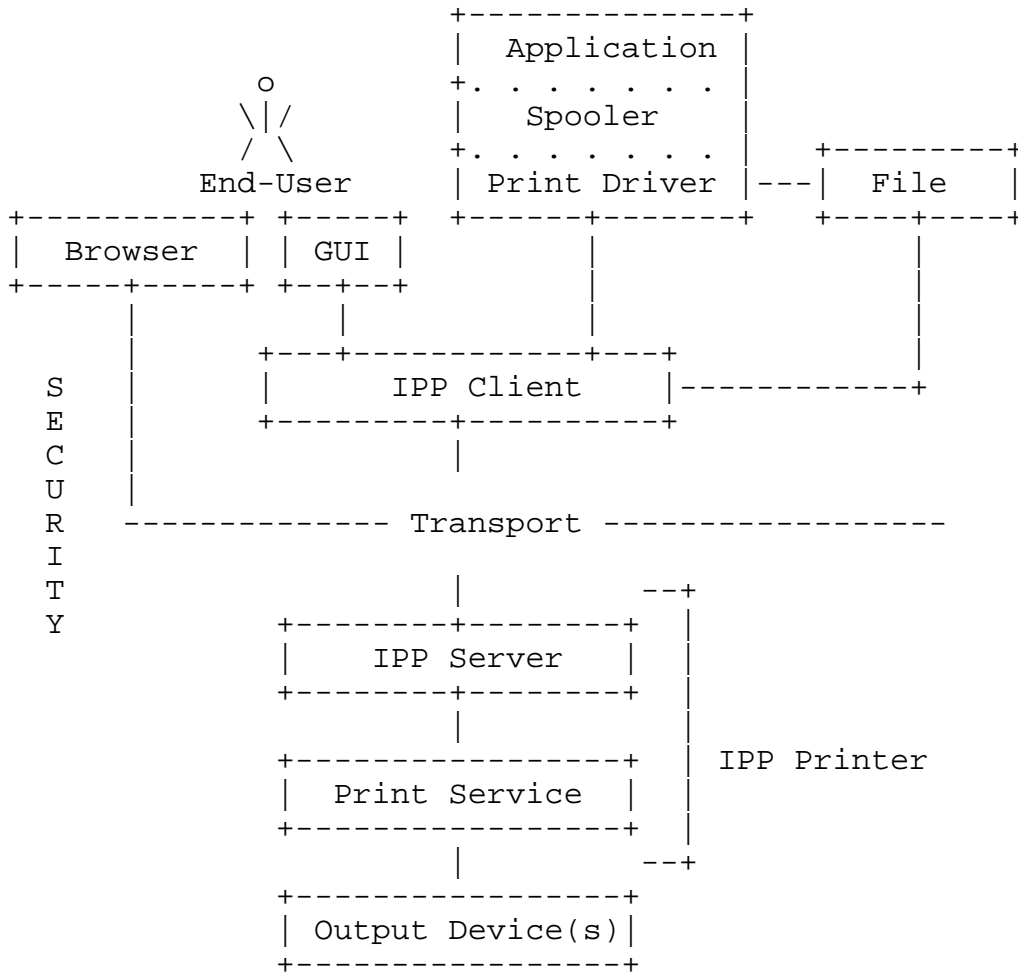
- 384 - Printer (Section 2.1)
- 385 - Job (Section 2.2)

386
387 Each object type has an associated set of operations (see section 3) and attributes (see section 4).

388 It is important, however, to understand that in real system implementations (which lie underneath the
389 abstracted IPP/1.0 model), there are other components of a print service which are not explicitly defined
390 in the IPP/1.0 model. The following figure illustrates where IPP/1.0 fits with respect to these other
391 components.

392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422

N D S
O I E
T R C
I E U
F C R
I T I
C O T
A R Y
T Y
I
O
N



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

IPP clients implement the IPP protocol on the client side, and give end users (or programs running on behalf of end users) the ability to query Printer objects and submit and manage print jobs. An IPP server is just that part of the Printer object that implements the server-side protocol. The rest of the Printer

435 object implements (or gateways into) the application semantics of the print service itself. The Printer
436 objects may be embedded in an output device or may be implemented on a host on the network that
437 communicates with an output device.

438 When a job is submitted to the Printer object and the Printer object validates the attributes in the
439 submission request, the Printer object creates a new Job object. The end user then interacts with this new
440 Job object to query its status and monitor the progress of the job. End users may also cancel the print job
441 by using the Job object's Cancel-Job operation. The notification service is out of scope for IPP/1.0, but
442 using such a notification service, the end user is able to register for and receive Printer specific and Job
443 specific events. An end user can query the status of Printer objects and can follow the progress of Job
444 objects by polling using the Get-Printer-Attributes, Get-Jobs, and Get-Job-Attributes operations.

445 2. IPP Objects

446 The IPP/1.0 model introduces objects of type Printer and Job. Each type of object models relevant
447 aspects of a real-world entity such as a real printer or real print job. Each object type is defined as a set
448 of possible attributes that may be supported by instances of that object type. For each object (instance),
449 the actual set of supported attributes and values describe a specific implementation. The object's
450 attributes and values describe its state, capabilities, realizable features, job processing functions, and
451 default behaviors and characteristics. For example, the Printer object type is defined as a set of attributes
452 that each Printer object potentially supports. In the same manner, the Job object type is defined as a set
453 of attributes that are potentially supported by each Job object.

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

455 - "MANDATORY": each object SHALL support the attribute.

456 - "OPTIONAL": each object MAY support the attribute.

457

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

460 2.1 Printer Object

461 The major component of the IPP/1.0 model is the Printer object. A Printer object implements the server-
462 side of the IPP/1.0 protocol. Using the protocol, end users may query the attributes of the Printer object
463 and submit print jobs to the Printer object. The actual implementation components behind the Printer
464 abstraction may take on different forms and different configurations. However, the model abstraction
465 allows the details of the configuration of real components to remain opaque to the end user. Section 3
466 describes each of the Printer operations in detail.

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

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

473

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

477 Some examples of configurations supporting a Printer object include:

- 478 1) An output device with no spooling capabilities
- 479 2) An output device with a built-in spooler
- 480 3) A print server supporting IPP with one or more associated output devices
 - 481 3a) The associated output devices may or may not be capable of spooling jobs
 - 482 3b) The associated output devices may or may not support IPP

483

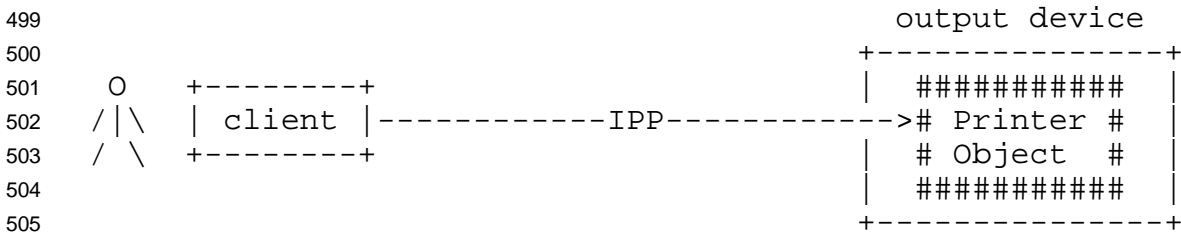
484 The following figures show some examples of how Printer objects can be realized on top of various
485 distributed printing configurations. The embedded case below represents configurations 1 and 2. The
486 hosted and fan-out figures below represent configurations 3a and 3b.

487 Legend:

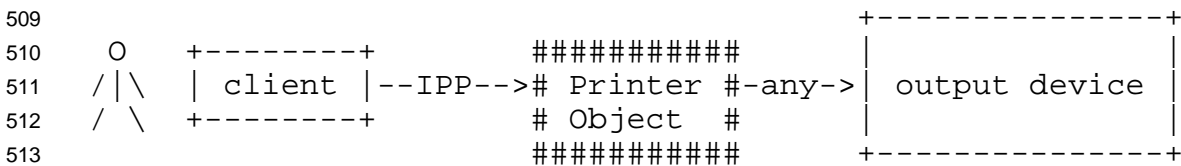
488
 489 ##### indicates a Printer object which is
 490 either embedded in an output device or is
 491 hosted in a server. The Printer object
 492 might or might not be capable of queuing/spooling.
 493

494 any indicates any network protocol or direct
 495 connect, including IPP
 496

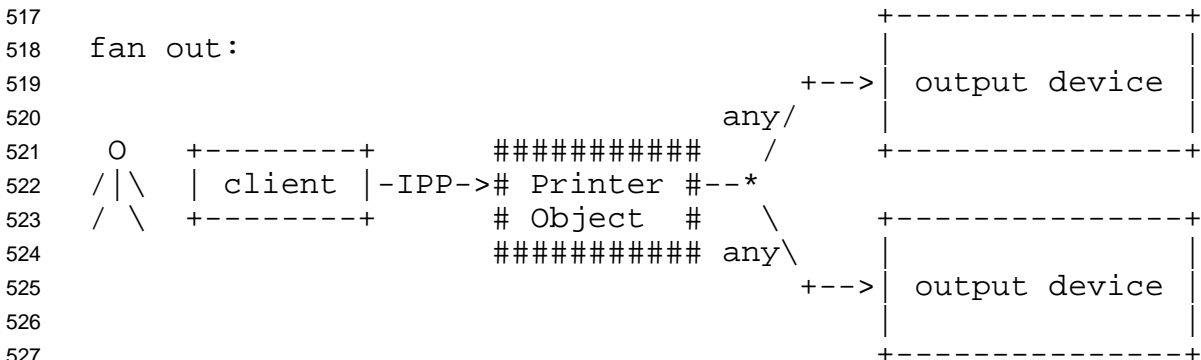
497
 498 embedded printer:



506
 507
 508 hosted printer:



514
 515
 516
 517 fan out:



528
 529
 530 2.2 Job Object

531 A Job object is used to model a print job. A Job can contain one or more documents. The information
532 required to create a Job object is sent in a create request from the end user via an IPP Client to the Printer
533 object. The Printer object validates the create request, and if the Printer object accepts the request, the
534 Printer object creates the new Job object. Section 3 describes each of the Job operations in detail.

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

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

543
544 A Job object contains at least one document, but may contain multiple documents. A document is either:

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

548
549 In IPP/1.0, a document is not modeled as an IPP object, therefore it has no object identifier or associated
550 attributes. All job processing instructions are modeled as Job object attributes. These attributes are
551 called Job Template attributes and they apply equally to all documents within a Job object.

552 2.3 Object Relationships

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

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

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

563 A Job object contains one or more documents. If the contained document is a stream of document data,
564 that stream can be contained in only one document. However, there can be identical copies of the stream
565 in other documents in the same or different Job objects. If the contained document is just a reference to a
566 stream of document data, other documents (in the same or different Job object(s)) may contain the same
567 reference.

568 2.4 Object Identity

569 All Printer and Job objects are identified by a Uniform Resource Identifier (URI) [RFC1630] so that they
570 can be persistently and unambiguously referenced. . The notion of a URI is a useful concept, however,
571 until the notion of URI is more stable (i.e., defined more completely and deployed more widely), it is
572 expected that the URIs used for IPP objects will actually be URLs [RFC1738] [RFC1808]. Since every
573 URL is a specialized form of a URI, even though the more generic term URI is used throughout the rest
574 of this document, its usage is intended to cover the more specific notion of URL as well.

575 An administrator configures Printer objects to either support or not support authentication and/or
576 message privacy using TLS [TLS] (the mechanism for security configuration is outside the scope of
577 IPP/1.0). In some situations, both types of connections (both authenticated and unauthenticated) can be
578 established using a single communication channel that has some sort of negotiation mechanism. In other
579 situations, multiple communication channels are used, one for each type of security configuration.
580 Section 8 provides a full description of all security considerations and configurations. ,

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

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

597 Note: IPP/1.0 does not specify how the client obtains the client supplied URI, but it is
598 RECOMMENDED that a Printer object be registered as an entry in a directory service. End-users and
599 programs can then interrogate the directory searching for Printers. Section 17 defines a generic schema
600 for Printer object entries in the directory service and describes how the entry acts as a bridge to the actual
601 IPP Printer object. The entry in the directory that represents the IPP Printer object includes the possibly
602 many URIs for that Printer object as values in one its attributes.

603 When a client submits a create request to the Printer object, the Printer object validates the request and
604 creates a new Job object. The Printer object assigns the new Job object a URI which is stored in the
605 "job-uri" Job attribute. This URI is then used by clients as the target for subsequent Job operations. The
606 Printer object generates a Job URI based on its configured security policy and the URI used by the client
607 in the create request.

608 For example, consider a Printer object that supports both a communication channel secured by the use of
609 TLS (using a standard URI indicating the use of HTTP over TLS) and another open communication
610 channel that is not secured with TLS (using an simple "http" schemed URI). If a client were to submit a
611 job using the secure URI, the Printer object would assign the new Job object a secure URI as well. If a
612 client were to submit a job using the open-channel URI, the Printer would assign the new Job object an
613 open-channel URI.

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

620 Allowing Job objects to have URIs allows for flexibility and scalability. For example, in some
621 implementations, the Printer object might create Jobs that are processed in the same local environment as
622 the Printer object itself. In this case, the Job URI might just be a composition of the Printer's URI and
623 some unique component for the Job object, such as the unique 32-bit positive integer mentioned later in
624 this paragraph. In other implementations, the Printer object might be a central clearing-house for
625 validating all Job object creation requests, but the Job object itself might be created in some environment
626 that is remote from the Printer object. In this case, the Job object's URI may have no physical-location
627 relationship at all to the Printer object's URI. Again, the fact that Job objects have URIs allows for
628 flexibility and scalability, however, many existing printing systems have local models or interface
629 constraints that force print jobs to be identified using only a 32-bit positive integer rather than an
630 independent URI. This numeric Job ID is only unique within the context of the Printer object to which
631 the create request was originally submitted. Therefore, in order to allow both types of client access to
632 IPP Job objects (either by Job URI or by numeric Job ID), when the Printer object successfully processes
633 a create request and creates a new Job object, the Printer object SHALL generate both a Job URI and a

634 Job ID. The Job ID (stored in the "job-id" attribute) only has meaning in the context of the Printer object
635 to which the create request was originally submitted. This requirement to support both Job URIs and Job
636 IDs allows all types of clients to access Printer objects and Job objects no matter the local constraints
637 imposed on the client implementation.

638 In addition to identifiers, Printer objects and Job objects have names. An object name need not be unique
639 across all instances of all objects. A Printer object's name is chosen and set by an administrator through
640 some mechanism outside the scope of IPP/1.0. A Job object's name is optionally chosen and supplied by
641 the IPP client submitting the job. If the client does not supply a Job object name, the Printer object
642 generates a name for the new Job object. In all cases, the name only has local meaning.

643 To summarize:

- 644 - Each Printer object is identified with one or more URIs. The Printer's "printer-uri-supported"
645 attribute contains the URI(s).
- 646 - The Printer object's "uri-security-supported" attribute identifies the communication channel security
647 protocols that may or may not have been configured for the various Printer object URIs (e.g., 'tls'
648 or 'none').
- 649 - Each Job object is identified with a Job URI. The Job's "job-uri" attribute contains the URI.
- 650 - Each Job object is also identified with Job ID which is a 32-bit, positive integer. The Job's "job-id"
651 attribute contains the Job ID. The Job ID is only unique within the context of the Printer object
652 which created the Job object.
- 653 - Each Job object has a "job-printer-uri" attribute which contains the URI of the Printer object that
654 was used to create the Job object. This attribute is used to determine the Printer object that
655 created a Job object when given only the URI for the Job object. This linkage is necessary to
656 determine the languages, charsets, and operations which are supported on that Job (the basis for
657 such support comes from the creating Printer object).
- 658 - Each Printer object has a name (which is not necessarily unique). The administrator chooses and
659 sets this name through some mechanism outside the scope of IPP/1.0 itself. The Printer object's
660 "printer-name" attribute contains the name.
- 661 - Each Job object has a name (which is not necessarily unique). The client optionally supplies this
662 name in the create request. If the client does not supply this name, the Printer object generates a
663 name for the Job object. The Job object's "job-name" attribute contains the name.

664 3. IPP Operations

665 IPP objects support operations. An operation consists of a request and a response. When a client
666 communicates with an IPP object, the client issues an operation request to the URI for that object.
667 Operations have attributes that supply information about the operation itself. These attributes are called
668 operation attributes (as compared to object attributes such as Printer object attributes or Job object

669 attributes). Each request carries along with it any operation attributes, object attributes, and/or document
670 data required to perform the operation. Each request requires a response from the object. Each response
671 indicates success or failure of the operation with a status code. The response contains any operation
672 attributes, object attributes, and/or status messages generated during the execution of the operation
673 request.

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

676 The IPP/1.0 Printer operations are:

- 677 Print-Job (section 3.2.1)
- 678 Print-URI (section 3.2.2)
- 679 Validate-Job (section 3.2.3)
- 680 Create-Job (section 3.2.4)
- 681 Get-Printer-Attributes (section 3.2.5)
- 682 Get-Jobs (section 3.2.6)

683

684 The Job operations are:

- 685 Send-Document (section 3.3.1)
- 686 Send-URI (section 3.3.2)
- 687 Cancel-Job (section 3.3.3)
- 688 Get-Job-Attributes (section 3.3.4)

689

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

692 3.1 Common Semantics

693 All IPP operations share some common elements and features. . These common elements are defined and
694 described in more detail in the following sections.

695 3.1.1 Required Elements

696 Every operation request contains:

- 697 - a "version-number",
698 - an "operation-id",
699 - a "request-id", and
700 - the attributes that are MANDATORY for that type of request.

701

702 Every operation response contains:

- 703 - a "version-number",
704 - a "status-code",
705 - the "request-id" that was supplied in the corresponding request, and
706 - the attributes that are MANDATORY for that type of response.

707

708 Note: The transport and encoding document [IPP-PRO] defines special rules for the encoding of the
709 "operation-id", the "version-number", the "status-code", and the "request-id". All other operation
710 elements represented using the more generic encoding rules for attributes and groups of attributes.

711

3.1.2 Operation IDs and Request IDs

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

715 In addition, every invocation of an operation is identified by a "request-id" value. For each request, the
716 client chooses the "request-id" which is an integer (possibly unique depending on client requirements) in
717 the range from 1 to $2^{*}31 - 1$ (inclusive). This "request-id" allows clients to manage multiple outstanding
718 requests. The receiving IPP object, copies the client supplied "request-id" attribute into the response so
719 that the client can match the response with the correct outstanding request.

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

725

3.1.3 Attributes

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

- 728 - Operation Attributes: These attributes are passed in the operation and affect the IPP object's
729 behavior while processing the operation request and may affect other attributes or groups of

730 attributes. Some operation attributes describe the document data associated with the print job
731 and are associated with new Job objects, however most operation attributes do not persist beyond
732 the life of the operation. The description of each operation attribute includes conformance
733 statements indicating which operation attributes are MANDATORY and which are OPTIONAL
734 for an IPP object to support and which attributes a client MUST supply in a request and an IPP
735 object MUST supply in a response.

- 736 - Job Template Attributes: These attributes affect the processing of a job. A client OPTIONALLY
737 supplies Job Template Attributes in a create request, and the receiving object MUST be prepared
738 to receive all supported attributes. The Job object can later be queried to find out what Job
739 Template attributes were originally requested in the create request, and such attributes are
740 returned in the response as Job Object Attributes. The Printer object can be queried about its Job
741 Template attributes to find out what type of job processing capabilities are supported and/or what
742 the default job processing behaviors are, though such attributes are returned in the response as
743 Printer Object Attributes. The "ipp-attribute-fidelity" operation attribute affects processing of all
744 client supplied Job Template attributes (see section 16 for a full description of "ipp-attribute-
745 fidelity" and its relationship to other attributes).
- 746 - Job Object Attributes: These attributes are returned in response to a query operation directed at a
747 Job object.
- 748 - Printer Object Attributes: These attributes are returned in response to a query operation directed at
749 a Printer object.
- 750 - Unsupported Attributes: In a create request, the client supplies a set of Operation and Job Template
751 attributes. If any of these attributes or their values are unsupported by the Printer object, the
752 Printer object returns the set of unsupported attributes in the response. Section 16 gives a full
753 description of how Job Template attributes supplied by the client in a create request are processed
754 by the Printer object and how unsupported attributes are returned to the client. Because of
755 extensibility, any IPP object might receive a request that contains new or unknown attributes or
756 values for which it has no support. In such cases, the IPP object processes what it can and returns
757 the unsupported attributes in the response.

758
759 Later in this section, each operation is formally defined by identifying the allowed and expected groups of
760 attributes for each request and response. The model identifies a specific order for each group in each
761 request or response, but the attributes within each group may be in any order, unless specified otherwise.

762 Each attribute specification includes the attribute's name followed by the name of its attribute syntax(es)
763 in parentheses. In addition, each 'integer' attribute is followed by the allowed range in parentheses,
764 (m:n), for values of that attribute. Each 'text' or 'name' attribute is followed by the maximum size in
765 octets in parentheses, (size), for values of that attribute. For more details on attribute syntax notation, see
766 the descriptions of these attributes syntaxes in section 4.1. It is an operational error for clients to supply
767 in operation requests and/or IPP objects to returns in operations responses attribute value(s) that do not

768 match the syntax(es) defined for that attribute (see section 3 for operation attributes and section 4 for IPP
769 object attributes).

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

774 Note: Some operations are MANDATORY for IPP objects to support; the others are OPTIONAL (see
775 section 5.2.2). Therefore, before using an OPTIONAL operation, a client SHOULD first use the
776 MANDATORY Get-Printer-Attributes operation to query the Printer's "operations-supported" attribute
777 in order to determine which OPTIONAL Printer and Job operations are actually supported. The client
778 SHOULD NOT use an OPTIONAL operation that is not supported. When an IPP object receives a
779 request to perform an operation it does not support, it returns the 'server-error-operation-not-supported'
780 status code (see section 14.1.5.2). An IPP object is non-conformant if it does not support a
781 MANDATORY operation.

782 3.1.4 Character Set and Natural Language Operation Attributes

783 Some Job and Printer attributes have values that are text strings and names intended for human
784 understanding rather than machine understanding (see the 'text' and 'name' attribute syntax descriptions in
785 section 4.1). The following sections describe two special Operation Attributes called "attributes-charset"
786 and "attributes-natural-language". These attributes are always part of the Operation Attributes group.
787 For most attribute groups, the order of the attributes within the group is not important. However, for
788 these two attributes within the Operation Attributes group, the order is critical. The "attributes-charset"
789 attribute MUST be the first attribute in the group and the "attributes-natural-language" attribute MUST
790 be the second attribute in the group. In other words, these attributes MUST be supplied in every IPP
791 request and response, they MUST come first in the group, and MUST come in the specified order. For
792 job creation operations, the IPP Printer implementation saves these two attributes with the new Job
793 object as Job Description attributes. For the sake of brevity in this document, these operation attribute
794 descriptions are not repeated with every operation request and response, but have a reference back to this
795 section instead.

796 3.1.4.1 Request Operation Attributes

797 The client SHALL supply and the Printer object SHALL support the following MANDATORY operation
798 attributes in every IPP/1.0 operation request:

799 "attributes-charset" (charset):

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

805
806 All IPP objects SHALL support the 'utf-8' charset [RFC2044] and MAY support additional
807 charsets provided that they are registered with IANA [IANA-CS]. If the Printer object does not
808 support the client supplied charset value, the Printer object SHALL reject the request and return
809 the 'client-error-charset-not-supported' status code. The Printer object SHALL indicate the
810 charset(s) supported as the values of the "charset-supported" Printer attribute (see Section
811 4.4.15), so that the client can query to determine which charset(s) are supported.

812
813 Note to client implementers: Since IPP objects are only required to support the 'utf-8' charset, in
814 order to maximize interoperability with multiple IPP object implementations, a client may want to
815 supply 'utf-8' in the "attributes-charset" operation attribute, even though the client is only passing
816 and able to present a simpler charset, such as US-ASCII or ISO-8859-1. Then the client will have
817 to filter out (or charset convert) those characters that are returned in the response that it cannot
818 present to its user. On the other hand, if both the client and the IPP objects also support a charset
819 in common besides utf-8, the client may want to use that charset in order to avoid charset
820 conversion or data loss.

821
822 See the 'charset' attribute syntax description in Section 4.1.7 for the syntax and semantic
823 interpretation of the values of this attribute and for example values.

824
825 "attributes-natural-language" (naturalLanguage):

826 This operation attribute identifies the natural language used by any 'text' and 'name' attributes that
827 the client is supplying in this request. This attribute also identifies the natural language that the
828 Printer object SHOULD use for all 'text' and 'name' attributes and status messages that the Printer
829 object returns in the response to this request.

830
831 There are no MANDATORY natural languages required for the Printer object to support.
832 However, the Printer object's "generated-natural-language-supported" attribute identifies the
833 natural languages supported by the Printer object and any contained Job objects for all text strings
834 generated by the IPP object. A client MAY query this attribute to determine which natural
835 language(s) are supported for generated messages.

836
837 For any of the attributes for which the Printer object generates text, i.e., for the "job-state-
838 message", "printer-state-message", and status messages (see Section 3.1.6), the Printer object

839 SHALL be able to generate these text strings in any of its supported natural languages. If the
840 client requests a natural language that is not supported, the Printer object SHALL return these
841 generated messages in the Printer's configured natural language as specified by the Printer's
842 "natural-language-configured" attribute" (see Section 4.4.16).

843
844 For other 'text' and 'name' attributes supplied by the client, authentication system, operator,
845 system administrator, or manufacturer, i.e., for "job-originating-user-name", "printer-name"
846 (name), "printer-location" (text), "printer-info" (text), and "printer-make-and-model" (text), the
847 Printer object is only required to support the configured natural language of the Printer identified
848 by the Printer object's "natural-language-configured" attribute, though support of additional
849 natural languages for these attributes is permitted.

850
851 For any 'text' or 'name' attribute in the request that is in a different natural language than the value
852 supplied in the "attributes-natural-language", the client SHALL use the Natural Language
853 Override mechanism (see sections 4.1.1.2 and 4.1.2.2) for each such attribute value supplied.

854
855 The IPP object SHALL accept any natural language and any Natural Language Override, whether
856 the IPP object supports that natural language or not (and independent of the value of the "ipp-
857 attribute-fidelity" Operation attribute). That is the IPP object accepts all client supplied values no
858 matter what the values are in the Printer object's "generated-natural-language-supported"
859 attribute. That attribute, "generated-natural-language-supported", only applies to generated
860 messages, not client supplied messages. The IPP object SHALL remember that natural language
861 for all client supplied attributes, and when returning those attributes in response to a query, the
862 IPP object SHALL indicate that natural language.

863
864 For example, the "job-name" attribute MAY be supplied by the client in a create request. The text
865 value for this attribute will be in the natural language identified by the "attribute-natural-language"
866 attribute, or if different, as identified by the Natural Language Override mechanism. If supplied,
867 the IPP object will use the value of the "job-name" attribute to populate the Job object's "job-
868 name" attribute. Whenever any client queries the Job object's "job-name" attribute, the IPP object
869 returns the attribute as stored and uses the Natural Language Override mechanism to specify the
870 natural language, if it is different from that reported in the "attributes-natural-language" operation
871 attribute of the response. An IPP object SHALL NOT reject a request based on a supplied
872 natural language in an "attributes-natural-language" Operation attribute or in any attribute that
873 uses the Natural Language Override.

874
875 See the 'naturalLanguage' attribute syntax description in section 4.1.8 for the syntax and semantic
876 interpretation of the values of this attribute and for example values.

877

878 Clients SHOULD NOT supply 'text' or 'name' attributes that use an illegal combination of natural
879 language and charset. For example, suppose a Printer object supports charsets 'utf-8', 'iso-8859-1', and
880 'iso-8859-7'. Suppose it also supports natural languages 'en' (English), 'fr' (French), and 'el' (Greek).
881 Although the Printer object supports the charset 'iso-8859-1' and natural language 'el', it probably does
882 not support the combination of Greek text strings using the 'iso-8859-1' charset. In a create request, if a
883 client supplies a "job-name" operation attribute that uses that specific invalid combination, it is a client
884 choice and it doesn't affect the Printer object or its correct operation to accept the invalid combination.
885 In this case, the Printer object simply accepts the client supplied value, stores it with the Job object, and
886 responds back with the same invalid combination whenever any client queries for that attribute. In a
887 query type operation (Get-Printer-Attributes for example), if the client requests an invalid combination,
888 the Printer object simply responds (as described below) using the Printer's configured natural language
889 rather than the natural language requested by the client. In either case, the Printer object does not reject
890 the request because of an invalid combination of charset and natural language (either at the global
891 operation level or at the Natural Language Override attribute-by-attribute level).

892 3.1.4.2 Response Operation Attributes

893 The Printer object SHALL supply and the client SHALL support the following MANDATORY operation
894 attributes in every IPP/1.0 operation response:

895 "attributes-charset" (charset):

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

901
902 If the Printer object supports more than just the 'utf-8' charset, the Printer object SHALL be able
903 to code convert between each of the charsets supported on a highest fidelity possible basis in
904 order to return the 'text' and 'name' attributes in the charset requested by the client. However,
905 some information loss MAY occur during the charset conversion depending on the charsets
906 involved. For example, the Printer object may convert from a UTF-8 'a' to a US-ASCII 'a' (with
907 no loss of information), from an ISO Latin 1 CAPITAL LETTER A WITH ACUTE ACCENT
908 to US-ASCII 'A' (losing the accent), or from a UTF-8 Japanese Kanji character to some ISO
909 Latin 1 error character indication such as '?', decimal code equivalent, or to the absence of a
910 character, depending on implementation.

911
912 Note: Whether an implementation that supports more than one charset stores the data in the
913 charset supplied by the client or code converts to one of the other supported charsets, depends on

914 implementation. The strategy should try to minimize loss of information during code conversion.
915 On each response, such an implementation converts from its internal charset to that requested.
916

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

918 This operation attribute identifies the natural language used by any 'text' and 'name' attributes that
919 the IPP object is returning in this response. Unlike the "attributes-charset" operation attribute, the
920 IPP object NEED NOT return the same value as that supplied by the client in the request. The
921 IPP object MAY return the natural language of the Job object or the Printer's configured natural
922 language as identified by the Printer object's "natural-language-configured" attribute, rather than
923 the natural language supplied by the client. For any 'text' or 'name' attribute or status message in
924 the response that is in a different natural language than the value returned in the "attributes-
925 natural-language" operation attribute, the IPP object SHALL use the Natural Language Override
926 mechanism (see sections 4.1.1.2 and 4.1.2.2) on each attribute value returned.

927 3.1.5 Operation Targets

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

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

- 934 - The Job object itself using the Job object's URI. In this case, the client identifies the target object by
935 supplying the correct URI in the "job-uri (uri)" operation attribute.
- 936 - The Printer object that created the Job object using both the Printer objects URI and the Job object's
937 Job ID. Since the Printer object that created the Job object generated the Job ID, it MUST be
938 able to correctly associate the client supplied Job ID with the correct Job object. The client
939 supplies the Printer object's URI in the "printer-uri (uri)" operation attribute and the Job object's
940 Job ID in the "job-id (integer(1:MAX))" operation attribute.

941
942 If the operation is directed at the Job object directly using the Job object's URI, the client SHALL NOT
943 include the redundant "job-id" operation attribute.

944 The operation target attributes are MANDATORY operation attributes that MUST be included in every
945 operation request. Like the charset and natural language attributes (see section 3.1.4), the operation
946 target attributes are specially ordered operation attributes. In all cases, the operation target attributes
947 immediately follow the "attributes-charset" and "attributes-natural-language" attributes within the
948 operation attribute group, however the specific ordering rules are :

- 949 - In the case where there is only one operation target attribute (i.e., either only the "printer-uri"
950 attribute or only the "job-uri" attribute), that attribute **MUST** be the third attribute in the
951 operation attributes group.
- 952 - In the case where Job operations use two operation target attributes (i.e., the "printer-uri" and "job-
953 id" attributes), the "printer-uri" attribute **MUST** be the third attribute and the "job-id" attribute
954 **MUST** be the fourth attribute.

955

956 Note: The IPP transport and encoding document [IPP-PRO] calls for the target URL to be included both
957 inside the IPP operation (as **MANDATORY** operation attributes) and outside the operation (at the HTTP
958 layer). The potential exists that these two values reference the same IPP object, but are not literally
959 identical since one can be a relative URL and the other can be an absolute URL. HTTP/1.1 allows clients
960 to generate and send a relative URL rather than an absolute URL. A relative URL identifies a resource
961 with the scope of the HTTP server, but does not include scheme, host or port. The following statements
962 characterize how URLs should be used in the mapping of IPP onto HTTP/1.1:

- 963 1. Although potentially redundant, a client **MUST** supply the target of the operation both as an
964 Operation Attribute (see Section 3.1.5) and as a URL at the HTTP layer. The rationale for this
965 decision is to maintain a consistent set of rules for mapping IPP to possibly many communication
966 layers, even where URLs are not used as the addressing mechanism.
- 967 2. Even though these two URLs might not be literally identical (one being relative and the other being
968 absolute), they must both reference the same IPP object.
- 969 3. The URL in the HTTP layer is either relative or absolute and is used by the HTTP server to route
970 the HTTP request to the correct resource relative to that HTTP server. The HTTP server need
971 not be aware of the URL within the operation request.
- 972 4. Once the HTTP server resource begins to process the HTTP request, it might get the reference to
973 the appropriate IPP Printer object from either the HTTP URL (using to the context of the HTTP
974 server for relative URLs) or from the URL within the operation request; the choice is up to the
975 implementation.
- 976 5. HTTP URLs can be relative or absolute, but the target URL in the operation **MUST** be an absolute
977 URL

978

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

- 980 1. If the URI scheme allows the port number to be explicitly included in the URI string, and a port
981 number is specified within the URI, then that port number **MUST** be used by the client to contact
982 the IPP object.
- 983
- 984 2. If the URI scheme allows the port number to be explicitly included in the URI string, and a port
985 number is not specified within the URI, then default port number implied by that URI **MUST** be
986 used by the client to contact the IPP object.

987

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

991

992 Note: The IPP transport and encoding document [IPP-PRO] shows a mapping of IPP onto HTTP/1.1
993 and defines a new default port number for using IPP over HTTP/1.1.

994

995 3.1.6 Operation Status Codes and Messages

996 Every operation response includes a MANDATORY "status-code" and an OPTIONAL "status-message"
997 operation attribute. The "status-code" provides information on the processing of a request. A "status-
998 message" attribute provides a short textual description of the status of the operation. The status code is
999 intended for use by automata, and the status message is intended for the human end user. If a response
1000 does include a "status-message" attribute, an IPP client NEED NOT examine or display the message,
1001 however it SHOULD do so in some implementation specific manner.

1002 The "status-code" value is a numeric value that has semantic meaning. The "status-code" syntax is
1003 similar to a "type2 enum" (see section 4.1 on "Attribute Syntaxes") except that values can range only
1004 from 0x0000 to 0x7FFF. Section 14 describes the status codes, assigns the numeric values, and suggests
1005 a corresponding status message for each status code. The "status-message" attribute's syntax is
1006 "text(255)".

1007 A client implementation of IPP SHOULD convert status code values into any localized message that has
1008 semantic meaning to the end user. If the Printer object supports the status message, the Printer object
1009 MUST be able to generate this message in any of the natural languages identified by the Printer object's
1010 "generated-natural-language-supported" attribute (see the "attributes-natural-language" operation
1011 attribute specified in section 3.1.4.1). As described in section 3.1.4.1 for any returned 'text' attribute, if
1012 there is a choice for generating this message, the Printer object uses the natural language indicated by the
1013 value of the "attributes-natural-language" in the client request if supported, otherwise the Printer object
1014 uses the value in the Printer object's own "natural-language-configured" attribute.

1015 3.1.7 Versions

1016 Each operation request and response carries with it a "version-number". Each value of the "version-
1017 number" is in the form "X.Y" where X is the major version number and Y is the minor version number.
1018 By including a version number in the client request, it allows the client to identify which version of IPP it

1019 is interested in using. If the IPP object does not support that version, the object responds with a status
1020 code of 'server-error-version-not-supported'.

1021 There is no version negotiation per se. However, if after receiving a 'server-error-version-not-supported'
1022 status code from an IPP object, there is nothing that prevents a client from trying again with a different
1023 version number. In order to conform to IPP/1.0, an implementation MUST support at least version '1.0'.

1024 There is only one notion of "version number" that covers both IPP Model and IPP Protocol changes.
1025 Thus the version number MUST change when introducing a new version of the Model document or a
1026 new version of the Protocol document.

1027 Changes to the major version number indicate structural or syntactic changes that make it impossible for
1028 older version of IPP clients and Printer objects to correctly parse and process the new or changed
1029 attributes, operations and responses. If the major version number changes, the minor version numbers is
1030 set to zero. As an example, adding the "ipp-attribute-fidelity" attribute (if it had not been part of version
1031 '1.0'), would have required a change to the major version number. Items that might affect the changing of
1032 the major version number include any changes to the protocol specification itself, such as:

- 1033 - reordering of ordered attributes or attribute sets
- 1034 - changes to the syntax of existing attributes
- 1035 - changing Operation or Job Template attributes from OPTIONAL to MANDATORY and vice versa
- 1036 - adding MANDATORY (for an IPP object to support) operation attributes
- 1037 - adding MANDATORY (for an IPP object to support) operation attribute groups
- 1038 - adding values to existing operation attributes
- 1039 - adding MANDATORY operations

1040
1041 Changes to the minor version number indicate the addition of new features, attributes and attribute values
1042 that may not be understood by all IPP objects, but which can be ignored if not understood. Items that
1043 might affect the changing of the minor version number include any changes to the model objects and
1044 attributes but not the protocol specification itself (except adding attribute syntaxes), such as:

- 1045 - grouping all extensions not included in a previous version into a new version
- 1046 - adding new attribute values
- 1047 - adding new object attributes
- 1048 - adding OPTIONAL (for an IPP object to support) operation attributes (i.e., those attributes that an
1049 IPP object can ignore without confusing clients)
- 1050 - adding OPTIONAL (for an IPP object to support) operation attribute groups (i.e., those attributes
1051 that an IPP object can ignore without confusing clients)
- 1052 - adding new attribute syntaxes
- 1053 - adding OPTIONAL operations

1054 - changing Job Description attributes or Printer Description attributes from OPTIONAL to
1055 MANDATORY or vice versa.

1056
1057 The encoding of the "operation-id", the "version-number", the "status-code", and the "request-id"
1058 SHALL NOT change over any version number (either major or minor). This rule guarantees that all
1059 future versions will be backwards compatible with all previous versions (at least for checking the
1060 "operation-id", the "version-number", and the "request-id"). In addition, any protocol elements
1061 (attributes, error codes, tags, etc.) that are not carried forward from one version to the next are
1062 deprecated so that they can never be reused with new semantics.

1063 Implementations that support a certain major version NEED NOT support ALL previous versions. As
1064 each new major version is defined (through the release of a new specification), that major version will
1065 specify which previous major versions MUST be supported in compliant implementations.

1066 3.1.8 Job Creation Operations

1067 In order to "submit a print job" and create a new Job object, a client issues a create request. A create
1068 request is any one of following three operation requests:

- 1069 - The Print-Job Request: A client that wants to submit a print job with only a single document uses
1070 the Print-Job operation. The operation allows for the client to "push" the document data to the
1071 Printer object by including the document data in the request itself.
1072
- 1073 - The Print-URI Request: A client that wants to submit a print job with only a single document
1074 (where the Printer object "pulls" the document data instead of the client "pushing" the data to the
1075 Printer object) uses the Print-URI operation. In this case, the client includes in the request only a
1076 URI reference to the document data (not the document data itself).
1077
- 1078 - The Create-Job Request: A client that wants to submit a print job with multiple documents uses the
1079 Create-Job operation. This operation is followed by an arbitrary number of Send-Document
1080 and/or Send-URI operations (each creating another document for the newly create Job object).
1081 The Send-Document operation includes the document data in the request (the client "pushes" the
1082 document data to the printer), and the Send-URI operation includes only a URI reference to the
1083 document data in the request (the Printer "pulls" the document data from the referenced location).
1084 The last Send-Document or Send-URI request for a given Job object includes a "last-document"
1085 operation attribute set to 'true' indicating that this is the last request.
1086

1087 Throughout this model specification, the term "create request" is used to refer to any of these three
1088 operation requests.

1089 A Create-Job operation followed by only one Send-Document operation is semantically equivalent to a
1090 Print-Job operation, however, for performance reasons, the client SHOULD use the Print-Job operation
1091 for all single document jobs. Also, Print-Job is a MANDATORY operation (all implementations MUST
1092 support it) whereas Create-Job is an OPTIONAL operation, hence some implementations might not
1093 support it.

1094 Job submission time is the point in time when a client issues a create request. The initial state of every
1095 Job object is the 'pending' or 'pending-held' state. Later, the Printer object begins processing the print job.
1096 At this point in time, the Job object's state moves to 'processing'. This is known as job processing time.
1097 There are validation checks that must be done at job submission time and others that must be performed
1098 at job processing time.

1099 At job submission time and at the time a Validate-Job operation is received, the Printer MUST do the
1100 following:

- 1101 1. Process the client supplied attributes and either accept or reject the request
- 1102 2. Validate the syntax of and support for the scheme of any client supplied URI

1103
1104 Section 16 describes the rules and issues surrounding the processing of client supplied attributes. Section
1105 16.3 presents suggested steps for an IPP object to either accept or reject any request. Section 16.4
1106 presents suggested additional steps for processing create requests.

1107 At job submission time the Printer SHOULD NOT perform the validation checks reserved for job
1108 processing time such as:

- 1109 1. Validating the document data
- 1110 2. Validating the actual contents of any client supplied URI (resolve the reference and follow the link
1111 to the document data)

1112
1113 At job submission time, these additional job processing time validation checks are essentially useless,
1114 since they require actually parsing and interpreting the document data, are not guaranteed to be 100%
1115 accurate, and MUST be done, yet again, at job processing time. Also, in the case of a URI, checking for
1116 availability at job submission time does not guarantee availability at job processing time. In addition, at
1117 job processing time, the Printer object might discover any of the following conditions that were not
1118 detectable at job submission time:

- 1119 - runtime errors in the document data,
- 1120 - nested document data that is in an unsupported format,
- 1121 - the URI reference is no longer valid (i.e., the server hosting the document might be down), or
- 1122 - any other job processing error

1123

1124 At job processing time, since the Printer object has already responded with a successful status code in the
1125 response to the create request, if the Printer object detects an error, the Printer object is unable to inform
1126 the end user of the error with an operation status code. In this case, the Printer, depending on the error,
1127 can set the "job-state", "job-state-reasons", or "job-state-message" attributes to the appropriate value(s)
1128 so that later queries can report the correct job status.

1129 Note: Asynchronous notification of events is outside the scope of IPP/1.0.

1130 3.2 Printer Operations

1131 All Printer operations are directed at Printer objects. A client **MUST** always supply the "printer-uri"
1132 operation attribute in order to identify the correct target of the operation.

1133 3.2.1 Print-Job Operation

1134 This **MANDATORY** operation allows a client to submit a print job with only one document and supply
1135 the document data (rather than just a reference to the data). See Section 16 for the suggested steps for
1136 processing create operations and their Operation and Job Template attributes.

1137 3.2.1.1 Print-Job Request

1138 The following groups of attributes are supplied as part of the Print-Job Request:

1139 Group 1: Operation Attributes

1140 Natural Language and Character Set:

1141 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1142 3.1.4.1. The Printer object **SHALL** copy these values to the corresponding Job Description
1143 attributes described in sections 4.3.23 and 4.3.24.

1144

1145 Target:

1146 The "printer-uri" (uri) operation attribute which is the target for this operation as described in
1147 section 3.1.5.

1148

1149 Requesting User Name:

1150 The "requesting-user-name" (name(MAX)) attribute **SHOULD** be supplied by the client as
1151 described in section 8.3.

1152

1153 "job-name" (name(MAX)):

1154 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1155 attribute. It contains the client supplied Job name. If this attribute is supplied by the client, its
1156 value is used for the "job-name" attribute of the newly created Job object. The client MAY
1157 automatically include any information that will help the end-user distinguish amongst his/her jobs,
1158 such as the name of the application program along with information from the document, such as
1159 the document name, document subject, or source file name. If this attribute is not supplied by the
1160 client, the Printer generates a name to use in the "job-name" attribute of the newly created Job
1161 object (see Section 4.3.5).

1162

1163 "ipp-attribute-fidelity" (boolean):

1164 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1165 attribute. The value 'true' indicates that total fidelity to client supplied Job Template attributes
1166 and values is required, else the Printer object SHALL reject the Print-Job request. The value
1167 'false' indicates that a reasonable attempt to print the Job object is acceptable and the Printer
1168 object SHALL accept the Print-job request. If not supplied, the Printer object assumes the value is
1169 'false'. All Printer objects MUST support both types of job processing. See section 16 for a full
1170 description of "ipp-attribute-fidelity" and its relationship to other attributes, especially the Printer
1171 object's "pdl-override-supported" attribute.

1172

1173 "document-name" (name(MAX)):

1174 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1175 attribute. It contains the client supplied document name. The document name MAY be different
1176 than the Job name. Typically, the client software automatically supplies the document name on
1177 behalf of the end user by using a file name or an application generated name. If this attribute is
1178 supplied, its value can be used in a manner defined by each implementation. Examples include:
1179 printed along with the Job (job start sheet, page adornments, etc.), used by accounting or
1180 resource tracking management tools, or even stored along with the document as a document level
1181 attribute. IPP/1.0 does not support the concept of document level attributes.

1182

1183 "document-format" (mimeMediaType) :

1184 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1185 attribute. The value of this attribute identifies the format of the supplied document data. If the
1186 client does not supply this attribute, the Printer object assumes that the document data is in the
1187 format defined by the Printer object's "document-format-default" attribute. If the client supplies
1188 this attribute, but the value is not supported by the Printer object, i.e., the value is not one of the
1189 values of the Printer object's "document-format-supported" attribute, the Printer object SHALL
1190 reject the request and return the 'client-error-document-format-not-supported' status code.

1191

1192 "document-natural-language" (naturalLanguage):

1193 The client **OPTIONALLY** supplies this attribute. The Printer object **OPTIONALLY** supports
1194 this attribute. This attribute specifies the natural language of the document for those document-
1195 formats that require a specification of the natural language in order to image the document
1196 unambiguously. There are no particular values required for the Printer object to support.

1197

1198 "compression" (type3 keyword)

1199 The client **OPTIONALLY** supplies this attribute. The Printer object **OPTIONALLY** supports
1200 this attribute and the "compression-supported" attribute (see section 4.4.29). The client supplied
1201 "compression" operation attribute identifies the compression algorithm used on the document
1202 data. If the client omits this attribute, the Printer object **SHALL** assume that the data is not
1203 compressed. If the client supplies the attribute and the Printer object supports the attribute, the
1204 Printer object uses the corresponding decompression algorithm on the document data. If the client
1205 supplies this attribute, but the value is not supported by the Printer object, i.e., the value is not
1206 one of the values of the Printer object's "compression-supported" attribute, the Printer object
1207 **SHALL** copy the attribute and its value to the Unsupported Attributes response group, reject the
1208 request, and return the 'client-error-attributes-or-values-not-supported' status code.

1209

1210 "job-k-octets" (integer(0:MAX))

1211 The client **OPTIONALLY** supplies this attribute. The Printer object **OPTIONALLY** supports
1212 this attribute and the "job-k-octets-supported" attribute (see section 4.4.30). The client supplied
1213 "job-k-octets" operation attribute identifies the total size of the document(s) in K octets being
1214 submitted (see section 4.3.17 for the complete semantics). If the client supplies the attribute and
1215 the Printer object supports the attribute, the value of the attribute is used to populate the Job
1216 object's "job-k-octets" Job Description attribute.

1217

1218 Note: For this attribute and the following two attributes ("job-impressions", and "job-media-
1219 sheets"), if the client supplies the attribute, but the Printer object does not support the attribute,
1220 the Printer object ignores the client-supplied value. If the client supplies the attribute and the
1221 Printer supports the attribute, and the value is within the range of the corresponding Printer
1222 object's "xxx-supported" attribute, the Printer object **SHALL** use the value to populate the Job
1223 object's "xxx" attribute. If the client supplies the attribute and the Printer supports the attribute,
1224 but the value is outside the range of the corresponding Printer object's "xxx-supported" attribute,
1225 the Printer object **SHALL** copy the attribute and its value to the Unsupported Attributes response
1226 group, reject the request, and return the 'client-error-attributes-or-values-not-supported' status
1227 code. If the client does not supply the attribute, the Printer object **MAY** choose to populate the
1228 corresponding Job object attribute depending on whether the Printer object supports the attribute
1229 and is able to calculate or discern the correct value.

1230

1231 "job-impressions" (integer(0:MAX))

1232 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports
1233 this attribute and the "job-impressions-supported" attribute (see section 4.4.31). The client
1234 supplied "job-impressions" operation attribute identifies the total size in number of impressions of
1235 the document(s) being submitted (see section 4.3.18 for the complete semantics).

1236

1237 See note under "job-k-octets".

1238

1239 "job-media-sheets" (integer(0:MAX))

1240 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports
1241 this attribute and the "job-media-sheets-supported" attribute (see section 4.4.32). The client
1242 supplied "job-media-sheets" operation attribute identifies the total number of media sheets to be
1243 produced for this job (see section 4.3.19 for the complete semantics).

1244

1245 See note under "job-k-octets".

1246

1247 Group 2: Job Template Attributes

1248 The client OPTIONALLY supplies a set of Job Template attributes as defined in section 4.2.

1249

1250 Group 3: Document Content

1251 The client MUST supply the document data to be processed.

1252

1253 Note: In addition to the MANDTORY common elements required for every operation request, the
1254 simplest Print-Job Request consists of just the "attributes-charset" and "attributes-natural-language"
1255 operation attributes; the "printer-uri" target operation attribute; the Document Content and and nothing
1256 else. In this simple case, the Printer object:

- 1257 - creates a new Job object (the Job object contains a single document),
- 1258 - stores a generated Job name in the "job-name" attribute in the natural language and charset
1259 requested (see Section 3.1.4.1) (if those are supported, otherwise using the Printer object's default
1260 natural language and charset), and
- 1261 - at job processing time, uses its corresponding default value attributes for the supported Job
1262 Template attributes that were not supplied by the client as IPP attribute or embedded instructions
1263 in the document data.

1264

1265 3.2.1.2 Print-Job Response

1266 The Printer object SHALL return to the client the following sets of attributes as part of the Print-Job
1267 Response:

1268 Group 1: Operation Attributes

1269 Status Message:

1270 In addition to the MANDATORY status code returned in every response, the response
1271 OPTIONALLY includes a "status-message" (text) operation attribute as described in section
1272 3.1.6. If the client supplies unsupported or conflicting Job Template attributes or values, the
1273 Printer object SHALL reject or accept the Print-Job request depending on the whether the client
1274 supplied a 'true' or 'false' value for the "ipp-attribute-fidelity" operation attribute. See section 16
1275 for a complete description of the suggested steps for processing a create request.
1276

1277 Natural Language and Character Set:

1278 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1279 3.1.4.2.
1280

1281 Group 2: Unsupported Attributes

1282 This is a set of Operation and Job Template attributes supplied by the client (in the request) that
1283 are not supported by the Printer object or that conflict with one another (see sections 16.3 and
1284 16.4).
1285

1286 Unsupported attributes fall into three categories:
1287

- 1288 1. The Printer object does not support the named attribute (no matter what the value).
- 1289 2. The Printer object does support the attribute, but does not support some or all of the particular
1290 values supplied by the client (i.e., the Printer object does not have those values in the
1291 corresponding supported values attribute).
- 1292 3. The Printer object does support the attributes and values supplied, but the particular values are
1293 in conflict with one another, because they violate a constraint, such as not being able to
1294 staple transparencies.
1295

1296 In the case of an unsupported attribute name, the Printer object returns the client-supplied
1297 attribute with a substituted "out-of-band" value of 'unsupported' indicating no support for the
1298 attribute itself (see the beginning of section 4.1).
1299

1300 In the case of a supported attribute with one or more unsupported values, the Printer object
1301 simply returns the client-supplied attribute with the unsupported values as supplied by the client.

1302 This indicates support for the attribute, but no support for that particular value. If the client
1303 supplies a multi-valued attribute with more than one value and the Printer object supports the
1304 attribute but only supports a subset of the client supplied values, the Printer object SHALL return
1305 only those values that are unsupported.

1306
1307 In the case of two (or more) supported attribute values that are in conflict with one another
1308 (although supported they values conflict when requested within the same job), the Printer object
1309 SHALL return all the values that it ignores or substitutes to resolve the conflict, but not any of the
1310 values that it is still using. The choice for exactly how to resolve the conflict is implementation
1311 dependent. See Section 16.4.4 for an example.

1312
1313 In these three cases, the value of the "ipp-attribute-fidelity" supplied by the client does not affect
1314 what the Printer object returns. The value of "ipp-attribute-fidelity" only affects whether the
1315 Print-Job operation is accepted or rejected. If the job is accepted, the client may query the job
1316 using the Get-Job-Attributes operation requesting the unsupported attributes that were returned in
1317 the create response to see which attributes were ignored (not stored on the Job object) and which
1318 attributes were stored with other (substituted) values.

1319
1320 Group 3: Job Object Attributes

1321 "job-uri" (uri):

1322 The Printer object MUST return the Job object's URI by returning the contents of the
1323 MANDATORY "job-uri" Job object attribute. The client uses the Job object's URI when
1324 directing operations at the Job object. The Printer object always uses its configured security
1325 policy when creating the new URI. However, if the Printer object supports more than one URI,
1326 the Printer object also uses information about which URI was used in the Print-Job Request to
1327 generated the new URI so that the new URI references the correct access channel. In other
1328 words, if the Print-Job Request comes in over a secure channel, the Printer object MUST generate
1329 a Job URI that uses the secure channel as well.

1330
1331 "job-id" (integer(1:MAX)):

1332 The Printer object MUST return the Job object's Job ID by returning the MANDATORY "job-
1333 id" Job object attribute. The client uses this "job-id" attribute in conjunction with the "printer-uri"
1334 attribute used in the Print-Job Request when directing Job operations at the Printer object.

1335
1336 "job-state":

1337 The Printer object MUST return the Job object's MANDATORY "job-state" attribute. The value
1338 of this attribute (along with the value of the next attribute "job-state-reasons") is taken from a
1339 "snapshot" of the new Job object at some meaningful point in time (implementation defined)

1340 between when the Printer object receives the Print-Job Request and when the Printer object
1341 returns the response.

1342

1343 "job-state-reasons":

1344 The Printer object OPTIONALLY returns the Job object's OPTIONAL "job-state-reasons"
1345 attribute. If the Printer object supports this attribute then it MUST be returned in the response.
1346 If this attribute is not returned in the response, the client can assume that the "job-state-reasons"
1347 attribute is not supported and will not be returned in a subsequent Job object query.

1348

1349 "job-state-message":

1350 The Printer object OPTIONALLY returns the Job object's OPTIONAL "job-state-message"
1351 attribute. If the Printer object supports this attribute then it MUST be returned in the response.
1352 If this attribute is not returned in the response, the client can assume that the "job-state-message"
1353 attribute is not supported and will not be returned in a subsequent Job object query.

1354

1355 "number-of-intervening-jobs":

1356 The Printer object OPTIONALLY returns the Job object's OPTIONAL "number-of-intervening-
1357 jobs" attribute. If the Printer object supports this attribute then it MUST be returned in the
1358 response. If this attribute is not returned in the response, the client can assume that the "number-
1359 of-intervening-jobs" attribute is not supported and will not be returned in a subsequent Job object
1360 query.

1361

1362 Note: Since any printer state information which affects a job's state is reflected in the "job-state"
1363 and "job-state-reasons" attributes, it is sufficient to return only these attributes and no specific
1364 printer status attributes.

1365

1366 Note: In addition to the MANDTORY common elements required for every operation response, the
1367 simplest response consists of the just the "attributes-charset" and "attributes-natural-language" operation
1368 attributes and the "job-uri", "job-id", and "job-state" Job Object Attributes. In this simplest case, the
1369 status code is "successful-ok" and there is no "status-message" operation attribute.

1370 3.2.2 Print-URI Operation

1371 This OPTIONAL operation is identical to the Print-Job operation (section 3.2.1) except that a client
1372 supplies a URI reference to the document data using the "document-uri" (uri) operation attribute (in
1373 Group 1) rather than including the document data itself. Before returning the response, the Printer
1374 MUST validate that the Printer supports the retrieval method (e.g., http, ftp, etc.) implied by the URI,
1375 and MUST check for valid URI syntax. If the client-supplied URI scheme is not supported, i.e. the value
1376 is not in the Printer object's "referenced-uri-scheme-supported" attribute, the Printer object SHALL reject
1377 the request and return the 'client-error-uri-scheme-not-supported' status code. See Section 16.3.5 for

1378 suggested additional checks. The Printer NEED NOT follow the reference and validate the contents of
1379 the reference.

1380 If the Printer object supports this operation, it MUST support the "reference-uri-schemes-supported"
1381 Printer attribute (see section 4.4.24).

1382 It is up to the IPP object to interpret the URI and subsequently "pull" the document from the source
1383 referenced by the URI string.

1384 3.2.3 Validate-Job Operation

1385 This MANDATORY operation is similar to the Print-Job operation (section 3.2.1) except that a client
1386 supplies no document data and the Printer allocates no resources (i.e., it does not create a new Job
1387 object). This operation is used only to verify capabilities of a printer object against whatever attributes
1388 are supplied by the client in the Validate-Job request. By using the Validate-Job operation a client can
1389 validate that an identical Print-Job operation (with the document data) would be accepted. The Validate-
1390 Job operation also performs the same security negotiation as the Print-Job operation (see section 8), so
1391 that a client can check that the client and Printer object security requirements can be met before
1392 performing a Print-Job operation.

1393 Note: The Validate-Job operation does not accept a "document-uri" attribute in order to allow a client to
1394 check that the same Print-URI operation will be accepted, since the client doesn't send the data with the
1395 Print-URI operation. The client SHOULD just issue the Print-URI request.

1396 The Printer object returns the same status codes, Operation Attributes (Group 1) and Unsupported
1397 Attributes (Group 2) as the Print-Job operation. However, no Job Object Attributes (Group 3) are
1398 returned, since no Job object is created.

1399 3.2.4 Create-Job Operation

1400 This OPTIONAL operation is similar to the Print-Job operation (section 3.2.1) except that in the Create-
1401 Job request, a client does not supply document data or any reference to document data. Also, the client
1402 does not supply any of the "document-name", "document-format", "compression", or "document-natural-
1403 language" operation attributes. This operation is followed by one or more Send-Document or Send-URI
1404 operations. In each of those operation requests, the client OPTIONALLY supplies the "document-
1405 name", "document-format", and "document-natural-language" attributes for each document in the multi-
1406 document Job object. If a Printer object supports the Create-Job operation, it MUST also support the
1407 Send-Document operation and also MAY support the Send-URI operation.

1408 3.2.5 Get-Printer-Attributes Operation

1409 This MANDATORY operation allows a client to request the values of the attributes of a Printer object.
1410 In the request, the client supplies the set of Printer attribute names and/or attribute group names in which
1411 the requester is interested. In the response, the Printer object returns a corresponding attribute set with
1412 the appropriate attribute values filled in.

1413 For Printer objects, the possible names of attribute groups are:

- 1414 - 'job-template': all of the Job Template attributes that apply to a Printer object (the last two columns
1415 of the table in Section 4.2).
- 1416 - 'printer-description': the attributes specified in Section 4.4.
- 1417 - 'all': the special group 'all' that includes all supported attributes.

1418
1419 Since a client MAY request specific attributes or named groups, there is a potential that there is some
1420 overlap. For example, if a client requests, 'printer-name' and 'all', the client is actually requesting the
1421 "printer-name" attribute twice: once by naming it explicitly, and once by inclusion in the 'all' group. In
1422 such cases, the Printer object NEED NOT return each attribute only once in the response even if it is
1423 requested multiple times. The client SHOULD NOT request the same attribute in multiple ways.

1424 It is NOT REQUIRED that a Printer object support all attributes belonging to a group (since some
1425 attributes are OPTIONAL). However, it is MANDATORY that each Printer object support all group
1426 names.

1427 3.2.5.1 Get-Printer-Attributes Request

1428 The following sets of attributes are part of the Get-Printer-Attributes Request:

1429 Group 1: Operation Attributes

1430 Natural Language and Character Set:

1431 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1432 3.1.4.1.

1433

1434 Target:

1435 The "printer-uri" (uri) operation attribute which is the target for this operation as described in
1436 section 3.1.5.

1437

1438 Requesting User Name:

1439 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
1440 described in section 8.3.

1441

1442 "requested-attributes" (1setOf keyword) :

1443 The client OPTIONALLY supplies a set of attribute names and/or attribute group names in whose
1444 values the requester is interested. The Printer object MUST support this attribute. If the client
1445 omits this attribute, the Printer SHALL respond as if this attribute had been supplied with a value
1446 of 'all'.

1447

1448 "document-format" (mimeMediaType) :

1449 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1450 attribute. This attribute is useful for a Printer object to determine the set of supported attribute
1451 values that relate to the requested document format. The Printer object SHALL return the
1452 attributes and values that it uses to validate a job on a create or Validate-Job operation in which
1453 this document format is supplied. The Printer object SHOULD return only (1) those attributes
1454 that are supported for the specified format and (2) the attribute values that are supported for the
1455 specified document format. By specifying the document format, the client can get the Printer
1456 object to eliminate the attributes and values that are not supported for a specific document format.
1457 For example, a Printer object might have multiple interpreters to support both
1458 'application/postscript' (for PostScript) and 'text/plain' (for text) documents. However, for only
1459 one of those interpreters might the Printer object be able to support "number-up" with values of
1460 '1', '2', and '4'. For the other interpreter it might be able to only support "number-up" with a value
1461 of '1'. Thus a client can use the Get-Printer-Attributes operation to obtain the attributes and values
1462 that will be used to accept/reject a create job operation.

1463

1464 Note: If the Printer object does not distinguish between different sets of supported values for
1465 each different document format when validating jobs in the create and Validate-Job operations, it
1466 SHALL NOT distinguish between different document formats in the Get-Printer-Attributes
1467 operation. If the Printer object does distinguish between different sets of supported values for
1468 each different document format specified by the client, this specialization applies only to the
1469 following Printer object attributes:

1470

- 1471 - Printer attributes that are Job Template attributes ("xxx-default" and xxx"-supported in the
- 1472 Table in Section 4.2),
- 1473 - "pdl-override-supported",
- 1474 - "compression-supported",
- 1475 - "job-k-octets-supported",
- 1476 - "job-impressions-supported",
- 1477 - "job-media-sheets-supported"
- 1478 - "printer-driver-installer",
- 1479 - "color-supported", and
- 1480 - "reference-uri-schemes-supported"

1481

1482 The values of all other Printer object attributes (including "document-format-supported") remain
1483 invariant with respect to the client supplied document format.

1484

1485 If the client omits this "document-format" operation attribute, the Printer object SHALL respond
1486 as if the attribute had been supplied with the value of the Printer object's "document-format-
1487 default" attribute. It is recommended that the client always supply a value for "document-format",
1488 since the Printer object's "document-format-default" may be 'application/octet-stream', in which
1489 case the returned attributes and values are for the union of the document formats that the Printer
1490 can automatically sense. For more details, see the description of the 'mimeType' attribute
1491 syntax in section 4.1.9.

1492

1493 If the client supplies a value for the "document-format" Operation attribute that is not supported
1494 by the Printer, i.e., is not among the values of the Printer object's "document-format-supported"
1495 attribute, the Printer object SHALL reject the operation and return the 'client-error-document-
1496 format-not-supported' status code.

1497

1498 3.2.5.2 Get-Printer-Attributes Response

1499 The Printer object returns the following sets of attributes as part of the Get-Printer-Attributes Response:

1500 Group 1: Operation Attributes

1501 Status Message:

1502 In addition to the MANDATORY status code returned in every response, the response
1503 OPTIONALLY includes a "status-message" (text) operation attribute as described in section
1504 3.1.5.

1505

1506 Natural Language and Character Set:

1507 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1508 3.1.4.2.

1509

1510 Group 2: Unsupported Attributes

1511 This is a set of Operation attributes supplied by the client (in the request) that are not supported
1512 by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16).

1513

1514 Group 3: Printer Object Attributes

1515 This is the set of requested attributes and their current values. The Printer object ignores (does
1516 not respond with) any requested attribute which is not supported. The Printer object MAY

1517 respond with a subset of the supported attributes and values, depending on the security policy in
1518 force. However, the Printer object SHALL respond with the 'unknown' value for any supported
1519 attribute (including all MANDATORY attributes) for which the Printer object does not know the
1520 value. Also the Printer object SHALL respond with the 'no-value' for any supported attribute
1521 (including all MANDATORY attributes) for which the system administrator has not configured a
1522 value. See the description of the "out-of-band" values in the beginning of Section 4.1.
1523

1524 3.2.6 Get-Jobs Operation

1525 This MANDATORY operation allows a client to retrieve the list of Job objects belonging to the target
1526 Printer object. The client may also supply a list of Job attribute names and/or attribute group names. A
1527 group of Job object attributes will be returned for each Job object that is returned.

1528 This operation is similar to the Get-Job-Attributes operation, except that this Get-Jobs operation returns
1529 attributes from possibly more than one object (see the description of Job attribute group names in section
1530 3.3.4).

1531 3.2.6.1 Get-Jobs Request

1532 The client submits the Get-Jobs request to a Printer object.

1533 The following groups of attributes are part of the Get-Jobs Request:

1534 Group 1: Operation Attributes

1535 Natural Language and Character Set:

1536 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1537 3.1.4.1.

1538

1539 Target:

1540 The "printer-uri" (uri) operation attribute which is the target for this operation as described in
1541 section 3.1.5.

1542

1543 Requesting User Name:

1544 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
1545 described in section 8.3.

1546

1547 "limit" (integer(1:MAX)):

1548 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1549 attribute. It is an integer value that indicates a limit to the number of Job objects returned. The

1550 limit is a "stateless limit" in that if the value supplied by the client is 'N', then only the first 'N' jobs
1551 are returned in the Get-Jobs Response. There is no mechanism to allow for the next 'M' jobs after
1552 the first 'N' jobs. If the client does not supply this attribute, the Printer object responds with all
1553 applicable jobs.

1554

1555 "requested-attributes" (1setOf keyword):

1556 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1557 attribute. It is a set of Job attribute names and/or attribute groups names in whose values the
1558 requester is interested. This set of attributes is returned for each Job object that is returned. The
1559 allowed attribute group names are the same as those defined in the Get-Job-Attributes operation
1560 in section 3.3.4. If the client does not supply this attribute, the Printer SHALL respond as if the
1561 client had supplied this attribute with two values: 'job-uri' and 'job-id'.

1562

1563 "which-jobs" (keyword):

1564 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1565 attribute. It indicates which Job objects SHALL be returned by the Printer object. The values for
1566 this attribute are:

1567

1568 'completed': This includes any Job object whose state is 'completed', 'canceled', or 'aborted'.

1569 'not-completed': This includes any Job object whose state is 'pending', 'processing',

1570 'processing-stopped', or 'pending-held'.

1571

1572 A Printer object SHALL support both values. However, if the implementation does not keep jobs
1573 in the 'completed', 'canceled', and 'aborted' states, then it returns no jobs when the 'completed'
1574 value is supplied.

1575

1576 If a client supplies some other value, the Printer object SHALL copy the attribute and the
1577 unsupported value to the Unsupported Attributes response group, reject the request, and return
1578 the 'client-error-attributes-or-values-not-supported' status code.

1579

1580 If the client does not supply this attribute, the Printer object SHALL respond as if the client had
1581 supplied the attribute with a value of 'not-completed'.

1582

1583 "my-jobs" (boolean):

1584 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1585 attribute. It indicates whether all jobs or just the jobs submitted by the requesting user of this
1586 request SHALL be returned by the Printer object. If the client does not supply this attribute, the
1587 Printer object SHALL respond as if the client had supplied the attribute with a value of 'false', i.e.,
1588 all jobs. The means for authenticating the requesting user and matching the jobs is described in
1589 section 8.

1590 3.2.6.2 Get-Jobs Response

1591 The Printer object returns all of the Job objects that match the criteria as defined by the attribute values
1592 supplied by the client in the request. It is possible that no Job objects are returned since there may
1593 literally be no Job objects at the Printer, or there may be no Job objects that match the criteria supplied by
1594 the client. If the client requests any Job attributes at all, there is a set of Job Object Attributes returned
1595 for each Job object.

1596 Group 1: Operation Attributes

1597 Status Message:

1598 In addition to the MANDATORY status code returned in every response, the response
1599 OPTIONALLY includes a "status-message" (text) operation attribute as described in section
1600 3.1.5.

1601

1602 Natural Language and Character Set:

1603 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1604 3.1.4.2.

1605

1606 Group 2: Unsupported Attributes

1607 This is a set of Operation attributes supplied by the client (in the request) that are not supported
1608 by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16.3).

1609

1610 Groups 3 to N: Job Object Attributes

1611 The Printer object responds with one set of Job Object Attributes for each returned Job object.
1612 The Printer object ignores (does not respond with) any requested attribute or value which is not
1613 supported or which is restricted by the security policy in force, including whether the requesting
1614 user is the user that submitted the job (job originating user) or not (see section 8). However, the
1615 Printer object SHALL respond with the 'unknown' value for any supported attribute (including all
1616 MANDATORY attributes) for which the Printer object does not know the value, unless it would
1617 violate the security policy. See the description of the "out-of-band" values in the beginning of
1618 Section 4.1.

1619

1620 For any job submitted in a different natural language than the natural language that the Printer
1621 object is returning in the "attributes-natural-language" operation attribute in the Get-Jobs
1622 response, the Printer SHALL indicate the submitted natural language by returning the Job object's
1623 "attributes-natural-language" as the first Job object attribute, which overrides the "attributes-
1624 natural-language" operation attribute value being returned by the Printer object. If any returned
1625 'text' or 'name' attribute includes a Natural Language Override as described in the sections 4.1.1.2

1626 and 4.1.2.2, the Natural Language Override overrides the Job object's "attributes-natural-
1627 language" value and/or the "attributes-natural-language" operation attribute value.

1628

1629 Jobs are returned in the following order:

- 1630 - If the client requests all 'completed' Jobs (Jobs in the 'completed', 'aborted', or 'canceled'
1631 states), then the Jobs are returned newest to oldest (with respect to actual completion
1632 time)
- 1633 - If the client requests all 'not-completed' Jobs (Jobs in the 'pending', 'processing', 'pending-
1634 held', and 'processing-stopped' states), then Jobs are returned in relative chronological
1635 order of expected time to complete (based on whatever scheduling algorithm is configured
1636 for the Printer object).
- 1637

1638 3.3 Job Operations

1639 All Job operations are directed at Job objects. A client **MUST** always supply some means of identifying
1640 the Job object in order to identify the correct target of the operation. That job identification **MAY** either
1641 be a single Job URI or a combination of a Printer URI with a Job ID. The IPP object implementation
1642 **MUST** support both forms of identification for every job.

1643 3.3.1 Send-Document Operation

1644 This **OPTIONAL** operation allows a client to create a multi-document Job object that is initially "empty"
1645 (contains no documents). In the Create-Job response, the Printer object returns the Job object's URI (the
1646 "job-uri" attribute) and the Job object's 32-bit identifier (the "job-id" attribute). For each new document
1647 that the client desires to add, the client uses a Send-Document operation. Each Send-Document Request
1648 contains the entire stream of document data for one document.

1649 Since the Create-Job and the send operations (Send-Document or Send-URI operations) that follow can
1650 occur over arbitrarily long periods of time, each Printer object must decide how long to "wait" for the
1651 next send operation. The Printer object **OPTIONALLY** supports the "multiple-operation-timeout"
1652 attribute. This attribute indicates the maximum number of seconds the Printer object will wait for the
1653 next send operation. If the Printer object times-out waiting for the next send operation, the Printer object
1654 **MAY** decide on any of the following semantic actions:

- 1655 1. Assume that the Job is an invalid job, start the process of changing the job state to 'aborted', and
1656 clean up all resources associated with the Job. In this case, if another send operation is finally
1657 received, the Printer responds with a "client-error-not-possible" or "client-error-not-found"
1658 depending on whether or not the Job object is still around when it finally arrives.

- 1659 2. Assume that the last send operation received was in fact the last document (as if the "last-
1660 document" flag had been set to 'true'), close the Job object, and proceed to process it (i.e., move
1661 the Job's state to 'pending').
- 1662 3. Assume that the last send operation received was in fact the last document, close the Job, but move
1663 it to the 'pending-held' to allow an operator to determine whether or not to continue processing
1664 the Job by moving it back to the 'pending' state.

1665

1666 Each implementation is free to decide the "best" action to take depending on local policy, the value of
1667 "ipp-attribute-fidelity", and/or any other piece of information available to it. If the choice is to abort the
1668 Job object, it is possible that the Job object may already have been processed to the point that some
1669 media sheet pages have been printed.

1670 3.3.1.1 Send-Document Request

1671 The following attribute sets are part of the Send-Document Request:

1672 Group 1: Operation Attributes

1673 Natural Language and Character Set:

1674 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1675 3.1.4.1.

1676

1677 Target:

1678 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation
1679 attribute(s) which define the target for this operation as described in section 3.1.5.

1680

1681 Requesting User Name:

1682 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
1683 described in section 8.3.

1684

1685 "document-name" (name(MAX)):

1686 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1687 attribute. It contains the client supplied document name. The document name MAY be different
1688 than the Job name. It might be helpful, but NEED NOT be unique across multiple documents in
1689 the same Job. Typically, the client software automatically supplies the document name on behalf
1690 of the end user by using a file name or an application generated name. See the description of the
1691 "document-name" operation attribute in the Print-Job Request (section 3.2.1.1) for more
1692 information about this attribute.

1693

1694 "document-format" (mimeMediaType) :

1695 The client OPTIONALLY supplies this attribute. The Printer object MUST support this
1696 attribute. The value of this attribute identifies the format of the supplied document data. If the
1697 client does not supply this attribute, the Printer object assumes that the document data is in the
1698 format defined by the Printer object's "document-format-default" attribute. If the client supplies
1699 this attribute, but the value is not supported by the Printer object, i.e., the value is not one of the
1700 values of the Printer object's "document-format-supported" attribute, the Printer object SHALL
1701 reject the request and return the 'client-error-document-format-not-supported' status code.

1702

1703 "document-natural-language" (naturalLanguage):

1704 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports
1705 this attribute. This attribute specifies the natural language of the document for those document-
1706 formats that require a specification of the natural language in order to image the document
1707 unambiguously. There are no particular values required for the Printer object to support.

1708

1709 "compression" (type3 keyword)

1710 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports
1711 this attribute and the "compression-supported" attribute (see section 4.4.29). The client supplied
1712 "compression" operation attribute identifies the compression algorithm used on the document
1713 data. If the client omits this attribute, the Printer object SHALL assume that the data is not
1714 compressed. If the client supplies the attribute and the Printer object supports the attribute, the
1715 Printer object SHALL use the corresponding decompression algorithm on the document data. If
1716 the client supplies this attribute, but the value is not supported by the Printer object, i.e., the value
1717 is not one of the values of the Printer object's "compression-supported" attribute, the Printer
1718 object SHALL copy the attribute and its value to the Unsupported Attributes response group,
1719 reject the request, and return the 'client-error-attributes-or-values-not-supported' status code.

1720

1721 "last-document" (boolean):

1722 The client MUST supply this attribute. The Printer object MUST support this attribute. It is a
1723 boolean flag that is set to 'true' if this is the last document for the Job, 'false' otherwise.

1724

1725 Group 2: Document Content

1726 The client MUST supply the document data if the "last-document" flag is set to 'false'. However,
1727 since a client might not know that the previous document sent with a Send-Document (or Send-
1728 URI) operation was the last document (i.e., the "last-document" attribute was set to 'false'), it is
1729 legal to send a Send-Document request with no document data where the "last-document" flag is
1730 set to 'true'. Such a request SHALL NOT increment the value of the Job object's "number-of-
1731 documents" attribute, since no real document was added to the job.

1732 3.3.1.2 Send-Document Response

1733 The following sets of attributes are part of the Send-Document Response:

1734 Group 1: Operation Attributes

1735 Status Message:

1736 In addition to the MANDATORY status code returned in every response, the response
1737 OPTIONALLY includes a "status-message" (text) operation attribute as described in section
1738 3.1.5.

1739

1740 Natural Language and Character Set:

1741 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1742 3.1.4.2.

1743

1744 Group 2: Unsupported Attributes

1745 This is a set of Operation attributes supplied by the client (in the request) that are not supported
1746 by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16.3).

1747

1748 Group 3: Job Object Attributes

1749 This is the same set of attributes as described in the Print-Job response (see section 3.2.1.2).

1750

1751 3.3.2 Send-URI Operation

1752 This OPTIONAL operation is identical to the Send-Document operation (see section 3.3.1) except that a
1753 client MUST supply a URI reference ("document-uri" operation attribute) rather than the document data
1754 itself. If a Printer object supports this operation, clients can use both Send-URI or Send-Document
1755 operations to add new documents to an existing multi-document Job object. However, if a client needs
1756 to indicate that the previous Send-URI or Send-Document was the last document, the client MUST use
1757 the Send-Document operation with no document data and the "last-document" flag set to 'true' (rather
1758 than using a Send-URI operation with no "document-uri" operation attribute). If a Printer object
1759 supports this operation, it MUST also support the Print-URI operation (see section 3.2.2).

1760 The Printer object MUST validate the syntax and URI scheme of the supplied URI before returning a
1761 response, just as in the Print-URI operation.

1762 3.3.3 Cancel-Job Operation

1763 This MANDATORY operation allows a client to cancel a Print Job any time after a create job operation.
1764 Since a Job might already be printing by the time a Cancel-Job is received, some media sheet pages might
1765 be printed before the job is actually terminated.

1766 3.3.3.1 Cancel-Job Request

1767 The following groups of attributes are part of the Cancel-Job Request:

1768 Group 1: Operation Attributes

1769 Natural Language and Character Set:

1770 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1771 3.1.4.1.

1772

1773 Target:

1774 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation
1775 attribute(s) which define the target for this operation as described in section 3.1.5.

1776

1777 Requesting User Name:

1778 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
1779 described in section 8.3.

1780

1781 "message" (text(127)):

1782 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports
1783 this attribute. It is a message to the operator. This "message" attribute is not the same as the "job-
1784 message-from-operator" attribute. That attribute is used to report a message from the operator to
1785 the end user that queries that attribute. This "message" operation attribute is used to send a
1786 message from the client to the operator along with the operation request. It is an implementation
1787 decision of how or where to display this message to the operator (if at all).

1788

1789 3.3.3.2 Cancel-Job Response

1790 The following sets of attributes are part of the Cancel-Job Response:

1791 Group 1: Operation Attributes

1792 Status Message:

1793 In addition to the MANDATORY status code returned in every response, the response
1794 OPTIONALLY includes a "status-message" (text) operation attribute as described in section
1795 3.1.5.

1796
1797 If the job is already in the 'completed', 'aborted', or 'canceled' state, or the 'process-to-stop-point'
1798 value is set in the Job's "job-state-reasons" attribute, the Printer object SHALL reject the request
1799 and return the 'client-error-not-possible' error status code.

1800
1801 Natural Language and Character Set:

1802 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1803 3.1.4.2.

1804
1805 Group 2: Unsupported Attributes

1806 This is a set of Operation attributes supplied by the client (in the request) that are not supported
1807 by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16.3).

1808

1809 Once a successful response has been sent, the implementation guarantees that the Job will eventually end
1810 up in the 'canceled' state. Between the time of the Cancel-Job operation is accepted and when the job
1811 enters the 'canceled' job-state (see section 4.3.7), the "job-state-reasons" attribute SHOULD contain the '
1812 processing-to-stop-point' value which indicates to later queries that although the Job might still be
1813 'processing', it will eventually end up in the 'canceled' state, not the 'completed' state.

1814 3.3.4 Get-Job-Attributes Operation

1815 This MANDATORY operation allows a client to request the values of attributes of a Job object and it is
1816 almost identical to the Get-Printer-Attributes operation (see section 3.2.5). The only differences are that
1817 the operation is directed at a Job object rather than a Printer object, there is no "document-format"
1818 operation attribute used when querying a Job object, and the returned attribute group is a set of Job
1819 object attributes rather than a set of Printer object attributes.

1820 For Jobs, the possible names of attribute groups are:

1821 - 'job-template': all of the Job Template attributes that apply to a Job object (the first column of the
1822 table in Section 4.2).

1823 - 'job-description': all of the Job Description attributes specified in Section 4.3.

1824 - 'all': the special group 'all' that includes all supported attributes.

1825

1826 Since a client MAY request specific attributes or named groups, there is a potential that there is some
1827 overlap. For example, if a client requests, 'job-name' and 'job-description', the client is actually requesting
1828 the "job-name" attribute once by naming it explicitly, and once by inclusion in the 'job-description' group.
1829 In such cases, the Printer object NEED NOT return the attribute only once in the response even if it is
1830 requested multiple times. The client SHOULD NOT request the same attribute in multiple ways.

1831 It is NOT REQUIRED that a Job object support all attributes belonging to a group (since some attributes
1832 are OPTIONAL). However it is MANDATORY that each Job object support all group names.

1833 3.3.4.1 Get-Job-Attributes Request

1834 The following groups of attributes are part of the Get-Job-Attributes Request when the request is
1835 directed at a Job object:

1836 Group 1: Operation Attributes

1837 Natural Language and Character Set:

1838 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1839 3.1.4.1.

1840

1841 Target:

1842 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation
1843 attribute(s) which define the target for this operation as described in section 3.1.5.

1844

1845 Requesting User Name:

1846 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
1847 described in section 8.3.

1848

1849 "requested-attributes" (1setOf keyword) :

1850 The client OPTIONALLY supplies this attribute. The IPP object MUST support this attribute.

1851 It is a set of attribute names and/or attribute group names in whose values the requester is
1852 interested. If the client omits this attribute, the IPP object SHALL respond as if this attribute had
1853 been supplied with a value of 'all'.

1854

1855 3.3.4.2 Get-Job-Attributes Response

1856 The Printer object returns the following sets of attributes as part of the Get-Job-Attributes Response:

1857 Group 1: Operation Attributes

1858 Status Message:

1859 In addition to the MANDATORY status code returned in every response, the response
1860 OPTIONALLY includes a "status-message" (text) operation attribute as described in section
1861 3.1.5.

1862

1863 Natural Language and Character Set:

1864 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1865 3.1.4.2. The "attributes-natural-language" MAY be the natural language of the Job object, rather
1866 than the one requested.

1867

1868 Group 2: Unsupported Attributes

1869 This is a set of Operation attributes supplied by the client (in the request) that are not supported
1870 by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16.3).

1871

1872 Group 3: Job Object Attributes

1873 This is the set of requested attributes and their current values. The IPP object ignores (does not
1874 respond with) any requested attribute or value which is not supported or which is restricted by the
1875 security policy in force, including whether the requesting user is the user that submitted the job
1876 (job originating user) or not (see section 8). However, the IPP object SHALL respond with the
1877 'unknown' value for any supported attribute (including all MANDATORY attributes) for which
1878 the IPP object does not know the value, unless it would violate the security policy. See the
1879 description of the "out-of-band" values in the beginning of Section 4.1.

1880 4. Object Attributes

1881 This section describes the attributes with their corresponding attribute syntaxes and values that are part of
1882 the IPP model. The sections below show the objects and their associated attributes which are included
1883 within the scope of this protocol. Many of these attributes are derived from other relevant specifications:

1884 - Document Printing Application (DPA) [ISO10175]

1885 - RFC 1759 Printer MIB [RFC1759]

1886

1887 Each attribute is uniquely identified in this document using a "keyword" (see section 13.2.1) which is the
1888 name of the attribute. The keyword is included in the section header describing that attribute.

1889 Note: Not only are keywords used to identify attributes, but one of the attribute syntaxes described
1890 below is "keyword" so that some attributes have keyword values. Therefore, these attributes are defined
1891 as having an attribute syntax that is a set of keywords.

1892 4.1 Attribute Syntaxes

1893 This section defines the basic attribute syntax types that all clients and IPP objects SHALL be able to
1894 accept in responses and accept in requests, respectively. Each attribute description in sections 3 and 4
1895 includes the name of attribute syntax(es) in the heading (in parentheses). A conforming implementation
1896 of an attribute SHALL include the semantics of the attribute syntax(es) so identified. Section 6.3
1897 describes how the protocol can be extended with new attribute syntaxes.

1898 The attribute syntaxes are specified in the following sub-sections, where the sub-section heading is the
1899 keyword name of the attribute syntax inside the single quotes. In operation requests and responses each
1900 attribute value MUST be represented as one of the attribute syntaxes specified in the sub-section heading
1901 for the attribute. In addition, the value of an attribute in a response (but not in a request) MAY be one of
1902 the "out-of-band" values. Standard "out-of-band" values are:

1903 'unknown': The attribute is supported by the IPP object, but the value is unknown to the IPP object
1904 for some reason.

1905 'unsupported': The attribute is unsupported by the IPP object. This value SHALL be returned only as
1906 the value of an attribute in the Unsupported Attributes Group.

1907 'no-value': The attribute is supported by the Printer object, but the system administrator has not yet
1908 configured a value.

1909

1910 The protocol specification defines mechanisms for passing "out-of-band" values. All attributes in a
1911 request SHALL have one or more values as defined in Sections 4.2 to 4.4. Thus clients SHALL NOT
1912 supply attributes with "out-of-band" values. All attribute in a response SHALL have one or more values
1913 as defined in Sections 4.2 to 4.4 or a single "out-of-band" value.

1914 Most attributes are defined to have a single attribute syntax. However, a few attributes (e.g., "job-sheet",
1915 "media", "job-hold-until") are defined to have several attribute syntaxes, depending on the value. These
1916 multiple attribute syntaxes are separated by the "|" character in the sub-section heading to indicate the
1917 choice. Since each value SHALL be tagged as to its attribute syntax in the protocol, a single-valued
1918 attribute instance may have any one of its attribute syntaxes and a multi-valued attribute instance may
1919 have a mixture of its defined attribute syntaxes.

1920 4.1.1 'text'

1921 A text attribute is an attribute whose value is sequence of zero or more characters encoded in a maximum
1922 of 1023 ('MAX') octets. MAX is the maximum length for all values of any text attribute. However, if an
1923 attribute will always contain values whose maximum length is much less than MAX, the definition of that
1924 attribute will include a qualifier that defines the maximum length for values of that attribute. For
1925 example: the "printer-location" attribute is specified as "printer-location (text(127))". In this case, text
1926 values for "printer-location" SHALL NOT exceed 127 octets; if supplied with a longer text string via
1927 some external interface, implementations are free to truncate to this shorter length limitation.

1928 In this specification, all text attributes are defined using the 'text' syntax. However, 'text' is used only for
1929 brevity; the formal interpretation of 'text' is: 'textWithoutLanguage | textWithLanguage'. That is, for any
1930 attribute defined in this specification using the 'text' attribute syntax, all IPP objects and clients SHALL
1931 accept, support, and return either the 'textWithoutLanguage' or 'textWithLanguage' attribute syntaxes in
1932 actual usage and protocol execution. The syntax 'text' never appears "on-the-wire".

1933 Both 'textWithoutLanguage' and 'textWithLanguage' are needed to support the real world needs of
1934 interoperability between sties and systems that use different natural languages as the basis for human
1935 communication. Generally, one natural language applies to all text attributes in a give request or
1936 response. The language is indicated by the "attributes-natural-language" operation attribute defined in
1937 section 3.1.4 or "attributes-natural-language" job attribute defined in section 4.3.24, and there is no need
1938 to identify the natural language for each text string on a value-by-value basis. In these cases, the attribute
1939 syntax 'textWithoutLanguage' is used for text attributes. In other cases, the client needs to supply or the
1940 Printer object needs to return a text value in a natural language that is different from the rest of the text
1941 values in the request or response. In these cases, the client or Printer object uses the attribute syntax
1942 'textWithLanguage' for text attributes (this is the Natural Language Override mechanism described in
1943 section 3.1.4).

1944 'textWithoutLanguage' and 'textWithLanguage' are described in more detail in the following sections.

1945 4.1.1.1 'textWithoutLanguage'

1946 The 'textWithoutLanguage' syntax indicates a value that is sequence of zero or more characters. Text
1947 strings are encoded using the rules of some charset. The Printer object SHALL support the UTF-8
1948 charset [RFC2044] and MAY support additional charsets to represent 'text' values, provided that the
1949 charsets are registered with IANA [IANA-CS]. See Section 4.1.7 for the specification of the 'charset'
1950 attribute syntax, including restricted semantics and examples of charsets.

1951 4.1.1.2 'textWithLanguage'

1952 The 'textWithLanguage' attribute syntax is a compound attribute syntax consisting of two parts: a
1953 'textWithoutLanguage' part plus an additional 'naturalLanguage' (see section 4.1.8) part that overrides the
1954 natural language in force. The 'naturalLanguage' part explicitly identifies the natural language that applies
1955 to the text part of that value and that value alone. For any give text attribute, the 'textWithoutLanguage'
1956 part is limited to the maximum length defined for that attribute, , but the 'naturalLanguage' part is always
1957 limited to 63 octets. Using the 'textWithLanguage' attribute syntax rather than the normal
1958 'textWithoutLanguage' syntax is the so-called Natural Language Override mechanism and **MUST** be
1959 supported by all IPP objects and clients.

1960

1961 If the attribute is multi-valued (1setOf text), then the 'textWithLanguage' attribute syntax **MUST** be used
1962 to explicitly specify each attribute value whose natural language needs to be overridden. Other values in
1963 a multi-valued 'text' attribute in a request or a response revert to the natural language of the operation
1964 attribute or to the "attributes-natural-language" Job attribute, if present, in the case of a Get-Jobs
1965 response.

1966

1967 In a create request, the Printer object **MUST** accept and store with the Job object any natural language in
1968 the "attributes-natural-language" operation attribute, whether the Printer object supports that natural
1969 language or not. Furthermore, the Printer object **MUST** accept and store any 'textWithLanguage'
1970 attribute value, whether the Printer object supports that natural language or not. These requirements are
1971 independent of the value of the "ipp-attribute-fidelity" operation attribute that the client **MAY** supply.

1972 Example: If the client supplies the "attributes-natural-language" operation attribute with the value: 'en'
1973 indicating English, but the value of the "job-name" attribute is in French, the client **MUST** use the
1974 'textWithLanguage' attribute syntax with the following two values:

1975 'fr': Natural Language Override indicating French

1976 'Rapport Mensuel': the job name in French

1977

1978 See the Protocol document [IPP-PRO] for a detailed example of the 'textWithLanguage' attribute syntax.

1979 4.1.2 'name'

1980 This syntax type is used for user-friendly strings, such as a Printer name, that, for humans, are more
1981 meaningful than identifiers. Names are usually never translated from one natural language to another.
1982 The 'name' attribute syntax is essentially the same as 'text', including the **MANDATORY** support of UTF-

1983 8 except that the sequence of characters is limited so that its encoded form SHALL NOT exceed 255
1984 octets.

1985
1986 Also like 'text', 'name' is really an abbreviated notation for either 'nameWithoutLanguage' or
1987 'nameWithLanguage'; all IPP objects and clients MUST support the notion of 'name' attributes using
1988 either the 'nameWithoutLanguage' or the 'nameWithLanguage' syntax during protocol execution.

1989 Note: Only the 'text' and 'name' attribute syntaxes permit the Natural Language Override mechanism.

1990 Some attributes are defined as 'type3 keyword | name'. These attributes support values that are either
1991 type3 keywords or names. This dual-syntax mechanism enables a site administrator to extend these
1992 attributes to legally include values that are locally defined by the site administrator. Such names are not
1993 registered with IANA.

1994 4.1.2.1 'nameWithoutLanguage'

1995 The 'nameWithoutLanguage' syntax indicates a value that is sequence of zero or more characters so that
1996 its encoded form does not exceed 127 octets.

1997 4.1.2.2 'nameWithLanguage'

1998 The 'nameWithLanguage' attribute syntax behaves the same as the 'textWithLanguage' syntax. If a name
1999 is in a language that is different than the rest of the object or operation, then this 'nameWithLanguage'
2000 syntax is used rather than the generic 'nameWithoutLanguage' syntax.

2001 The 'nameWithLanguage' attribute syntax is a compound attribute syntax consisting of two parts: a
2002 'nameWithoutLanguage' part plus an additional 'naturalLanguage' (see section 4.1.8) part that overrides
2003 the natural language in force. The 'naturalLanguage' part explicitly identifies the natural language that
2004 applies to the that name value and that name value alone.

2005 Example: If the client supplies the "attributes-natural-language" operation attribute with the value: 'en'
2006 indicating English, but the "printer-name" attribute is in German, the client MUST use the
2007 'nameWithLanguage' attribute syntax as follows:

2008 'de': Natural Language Override indicating German

2009 'Farbdrucker': the Printer name in German

2010

2011 4.1.3 'keyword'

2012 The 'keyword' attribute syntax is a sequence of characters, length: 1 to 255, containing only the US-
2013 ASCII [ASCII] encoded values for lowercase letters ("a" - "z"), digits ("0" - "9"), hyphen ("-"), dot ("."),
2014 and underscore ("_"). The first character MUST be a lowercase letter. Furthermore, keywords SHALL
2015 be in U.S. English.

2016 This syntax type is used for enumerating semantic identifiers of entities in the abstract protocol, i.e.,
2017 entities identified in this document. Keywords are used as attribute names or values of attributes. Unlike
2018 'text' and 'name' attribute values, 'keyword' values SHALL NOT use the Natural Language Override
2019 mechanism, since they SHALL always be US-ASCII and U.S. English.

2020 Keywords are for use in the protocol. A user interface will likely provide a mapping between protocol
2021 keywords and displayable user-friendly words and phrases which are localized to the natural language of
2022 the user. While the keywords specified in this document MAY be displayed to users whose natural
2023 language is U.S. English, they MAY be mapped to other U.S. English words for U.S. English users, since
2024 the user interface is outside the scope of this document.

2025 In the definition for each attribute of this syntax type, the full set of defined keyword values for that
2026 attribute are listed.

2027 When a keyword is used to represent an attribute (its name), it MUST be unique within the full scope of
2028 all IPP objects and attributes. When a keyword is used to represent a value of an attribute, it MUST be
2029 unique just within the scope of that attribute. That is, the same keyword SHALL NOT be used for two
2030 different values within the same attribute to mean two different semantic ideas. However, the same
2031 keyword MAY be used across two or more attributes, representing different semantic ideas for each
2032 attribute. Section 6.1 describes how the protocol can be extended with new keyword values. Examples
2033 of attribute name keywords:

2034 "job-name"
2035 "attributes-charset"
2036

2037 Note: This document uses "type1", "type2", and "type3" prefixes to the "keyword" and "enum" basic
2038 syntaxes. This extra information applies only to how the set of values defined for attributes with these
2039 syntaxes can be extended; this extra information is not carried in the protocol itself. "type1" indicates
2040 that new versions of the IPP standards documents must be revised and issued in order for new values to
2041 be added. "type2" indicates that IPP Subject Matter Experts must work with IANA to review and
2042 approve any proposed new values before the new values can be registered. "type3" indicates that IPP
2043 Subject Matter Experts are not required to review and approve any proposed new values before the new
2044 values can be registered with IANA. These extensibility mechanisms and restrictions are fully described
2045 in section 6.1.

2046 4.1.4 'enum'

2047 The 'enum' attribute syntax is an enumerated integer value that is in the range from 1 to 2**31 - 1
2048 (MAX). Each value has an associated 'keyword' name. In the definition for each attribute of this syntax
2049 type, the full set of possible values for that attribute are listed. This syntax type is used for attributes for
2050 which there are enum values assigned by other standards, such as SNMP MIBs. A number of attribute
2051 enum values in this specification are also used for corresponding attributes in other standards [RFC1759].
2052 This syntax type is not used for attributes to which the system administrator may assign values. Section
2053 6.1 describes how the protocol can be extended with new enum values.

2054 Enum values are for use in the protocol. A user interface will provide a mapping between protocol enum
2055 values and displayable user-friendly words and phrases which are localized to the natural language of the
2056 user. While the enum symbols specified in this document MAY be displayed to users whose natural
2057 language is U.S. English, they MAY be mapped to other U.S. English words for U.S. English users, since
2058 the user interface is outside the scope of this document.

2059 Note: SNMP MIBs use '2' for 'unknown' which corresponds to the IPP "out-of-band" value 'unknown'.
2060 See the description of the "out-of-band" values at the beginning of Section 4.1. Therefore, attributes of
2061 type 'enum' start at '3'.

2062 4.1.5 'uri'

2063 The 'uri' attribute syntax is any valid Uniform Resource Identifier or URI [RFC1630]. Most often, URIs
2064 are simply Uniform Resource Locators or URLs [RFC1738] [RFC1808]. The maximum length of URIs
2065 used within IPP is 1023 octets. Although most other IPP syntax types allow for only lower-cased values,
2066 this syntax type allows for mixed-case values. The URI and URL standards allow for mixed-case values
2067 that are case-sensitive.

2068 4.1.6 'uriScheme'

2069 The 'uriScheme' attribute syntax is a sequence of characters representing a URI scheme according to RFC
2070 1738 [RFC1738]. Though RFC 1736 requires that the values be case-insensitive, IPP requires all lower
2071 case to simplify comparing by IPP clients and Printer objects. Standard values for this syntax type are the
2072 following keywords:

2073 'http': for HTTP schemed URIs (e.g., "http:...")
2074 'https': for use with non-standard HTTPS schemed URIs (e.g., "https:...")
2075 'ftp': for FTP schemed URIs (e.g., "ftp:...")
2076 'mailto': for SMTP schemed URIs (e.g., "mailto:...")
2077 'file': for file schemed URIs (e.g., "file:...")
2078

2079 A Printer object MAY support any URI scheme that has been registered with IANA [IANA-MT]. The
2080 maximum length of URI schemes used within IPP is 63 octets.

2081 4.1.7 'charset'

2082 The 'charset' attribute syntax is a standard identifier for a charset. A charset is a coded character set and
2083 encoding scheme. Charsets are used for labeling certain document contents and 'text' and 'name' attribute
2084 values. The syntax and semantics of this attribute syntax are specified in RFC 2046 [RFC2046] and
2085 contained in the IANA character-set Registry [IANA-CS] according to the IANA procedures [IANA-
2086 CSa]. Though RFC 2046 requires that the values be case-insensitive US-ASCII, IPP requires all lower
2087 case to simplify comparing by IPP clients and Printer objects. When a character-set in the IANA registry
2088 has more than one name (alias), the name labeled as "(preferred MIME name)", if present, SHALL be
2089 used.

2090 The maximum length of charset values used within IPP is 63 octets.

2091 Some examples are:

2092 'utf-8': ISO 10646 Universal Multiple-Octet Coded Character Set (UCS) represented as the UTF-8
2093 [RFC2279] transfer encoding scheme in which US-ASCII is a subset charset.

2094 'us-ascii': 7-bit American Standard Code for Information Interchange (ASCII), ANSI X3.4-1986
2095 [ASCII]. That standard defines US-ASCII, but RFC 2045 [46] eliminates most of the control
2096 characters from conformant usage in MIME and IPP.

2097 'iso-8859-1': 8-bit One-Byte Coded Character Set, Latin Alphabet Nr 1 [ISO8859-1]. That standard
2098 defines a coded character set that is used by Latin languages in the Western Hemisphere and
2099 Western Europe. US-ASCII is a subset charset.

2100 'iso-10646-ucs-2': ISO 10646 Universal Multiple-Octet Coded Character Set (UCS) represented as
2101 two octets (UCS-2), with the high order octet of each pair coming first (so-called Big Endian
2102 integer).

2103

2104 Some attribute descriptions MAY place additional requirements on charset values that may be used, such
2105 as MANDATORY values that MUST be supported or additional restrictions, such as requiring that the
2106 charset have US-ASCII as a subset charset.

2107 4.1.8 'naturalLanguage'

2108 The 'naturalLanguage' attribute syntax is a standard identifier for a natural language and optionally a
2109 country. The values for this syntax type are defined by RFC 1766 [RFC1766]. Though RFC 1766
2110 requires that the values be case-insensitive US-ASCII, IPP requires all lower case to simplify comparing
2111 by IPP clients and Printer objects. Examples include:

2112 'en': for English
2113 'en-us': for US English
2114 'fr': for French
2115 'de': for German
2116
2117 The maximum length of naturalLanguage values used within IPP is 63 octets.

2118 4.1.9 'mimeMediaType'

2119 The 'mimeMediaType' attribute syntax is the Internet Media Type (sometimes called MIME type) as
2120 defined by RFC 2046 [RFC2046] and registered according to the procedures of RFC 2048 [RFC2048]
2121 for identifying a document format. The value MAY include a charset parameter, depending on the
2122 specification of the Media Type in the IANA Registry [IANA-MT]. Although most other IPP syntax
2123 types allow for only lower-cased values, this syntax type allows for mixed-case values.

2124 Examples are:

2125 'text/html': An HTML document
2126 'text/plain': A plain text document in US-ASCII (RFC 2046 indicates that in the absence of the
2127 charset parameter SHALL mean US-ASCII rather than simply unspecified) [RFC2046].
2128 'text/plain; charset=US-ASCII': A plain text document in US-ASCII [52, 56].
2129 'text/plain; charset=ISO-8859-1': A plain text document in ISO 8859-1 (Latin 1) [ISO8859-1].
2130 'text/plain; charset=utf-8': A plain text document in ISO 10646 represented as UTF-8 [RFC2044]
2131 'text/plain, charset=iso-10646-ucs-2': A plain text document in ISO 10646 represented in two octets
2132 (UCS-2) [ISO10646-1]
2133 'application/postscript': A PostScript document [RFC2046]
2134 'application/vnd.hp-PCL': A PCL document [IANA-MT] (charset escape sequence embedded in the
2135 document data)
2136 'application/octet-stream': Auto-sense - see below

2137
2138 One special type is 'application/octet-stream'. If the Printer object supports this value, the Printer object
2139 SHALL be capable of auto-sensing the format of the document data. If the Printer object's default value
2140 attribute "document-format-default" is set to 'application/octet-stream', the Printer object not only
2141 supports auto-sensing of the document format, but will depend on the result of applying its auto-sensing
2142 when the client does not supply the "document-format" attribute. If the client supplies a document
2143 format value, the Printer SHALL rely on the supplied attribute, rather than trust its auto-sensing
2144 algorithm. To summarize:

2145 1. If the client does not supply a document format value, the Printer MUST rely on its default value
2146 setting (which may be 'application/octet-stream' indicating an auto-sensing mechanism).

- 2147 2. If the client supplies a value other than 'application/octet-stream', the client is supplying valid
2148 information about the format of the document data and the Printer object SHALL trust the client
2149 supplied value more than the outcome of applying an automatic format detection mechanism. For
2150 example, the client may be requesting the printing of a PostScript file as a 'text/plain' document.
2151 The Printer object SHALL print a text representation of the PostScript commands rather than
2152 interpret the stream of PostScript commands and print the result.
- 2153 3. If the client supplies a value of 'application/octet-stream', the client is indicating that the Printer
2154 object SHALL use its auto-sensing mechanism on the client supplied document data whether
2155 auto-sensing is the Printer object's default or not.

2156

2157 Note: Since the auto-sensing algorithm is probabilistic, if the client requests both auto-sensing
2158 ("document-format" set to 'application/octet-stream') and true fidelity ("ipp-attribute-fidelity" set to
2159 'true'), the Printer object might not be able to guarantee exactly what the end user intended (the auto-
2160 sensing algorithm might mistake one document format for another), but it is able to guarantee that its
2161 auto-sensing mechanism be used.

2162 The maximum length of a 'mimeType' value in IPP is 255 octets.

2163 4.1.10 'octetString'

2164 The 'octetString' attribute syntax is a sequence of octets encoded in a maximum of 1023 octets which is
2165 indicated in sub-section headers using the notation: octetString(MAX). This syntax type is used for
2166 opaque data.

2167 4.1.11 'boolean'

2168 The 'boolean' attribute syntax is similar to an enum with only two values: 'true' and 'false'.

2169 4.1.12 'integer'

2170 The 'integer' attribute syntax is an integer value that is in the range from -2^{31} (MIN) to $2^{31} - 1$
2171 (MAX). Each individual attribute may specify the range constraint explicitly in sub-section headers if the
2172 range is different from the full range of possible integer values. For example: job-priority
2173 (integer(1:100)) for the "job-priority" attribute. However, the enforcement of that additional constraint is
2174 up to the IPP objects, not the protocol.

2175 4.1.13 'rangeOfInteger'

2176 The 'rangeOfInteger' attribute syntax is an ordered pair of integers that defines an inclusive range of
2177 integer values. The first integer specifies the lower bound and the second specifies the upper bound. If a

2178 range constraint is specified in the header description for an attribute in this document whose attribute
2179 syntax is 'rangeOfInteger' (i.e., 'X:Y' indicating X as a minimum value and Y as a maximum value), then
2180 the constraint applies to both integers.

2181 4.1.14 'dateTime'

2182 The 'dateTime' attribute syntax is a standard, fixed length, 11 octet representation of the "DateAndTime"
2183 syntax as defined in RFC 1903 [RFC1903]. RFC 1903 also identifies an 8 octet representation of a
2184 "DateAndTime" value, but IPP objects MUST use the 11 octet representation. A user interface will
2185 provide a mapping between protocol dateTime values and displayable user-friendly words or presentation
2186 values and phrases which are localized to the natural language and date format of the user.

2187 4.1.15 'resolution'

2188 The 'resolution' attribute syntax specifies a two-dimensional resolution in the indicated units. It consists
2189 of 3 integers: a cross feed direction resolution (positive integer value), a feed direction resolution
2190 (positive integer value), and a units value. The semantics of these three components are taken from the
2191 Printer MIB [RFC1759] suggested values. That is, the cross feed direction component resolution
2192 component is the same as the prtMarkerAddressabilityXFeedDir object in the Printer MIB, the feed
2193 direction component resolution component is the same as the prtMarkerAddressabilityFeedDir in the
2194 Printer MIB, and the units component is the same as the prtMarkerAddressabilityUnit object in the
2195 Printer MIB (namely, '3' indicates dots per inch and '4' indicates dots per centimeter). All three values
2196 MUST be present even if the first two values are the same. Example: '300', '600', '3' indicates a 300 dpi
2197 cross-feed direction resolution, a 600 dpi feed direction resolution, since a '3' indicates dots per inch
2198 (dpi).

2199 4.1.16 '1setOf X'

2200 The '1setOf X' attribute syntax is 1 or more values of attribute syntax type X. This syntax type is used
2201 for multi-valued attributes. The syntax type is called '1setOf' rather than just 'setOf' as a reminder that
2202 the set of values SHALL NOT be empty (i.e., a set of size 0). Sets are normally unordered. However
2203 each attribute description of this type may specify that the values MUST be in a certain order for that
2204 attribute.

2205 4.2 Job Template Attributes

2206 Job Template attributes describe job processing behavior. Support for Job Template attributes by a
2207 Printer object is OPTIONAL (see section 13.2.3 for a description of support for OPTIONAL attributes).
2208 Also, clients OPTIONALLY supply Job Template attributes in create requests.

2209 Job Template attributes conform to the following rules. For each Job Template attribute called "xxx":

2210 1. If the Printer object supports "xxx" then it SHALL support both a "xxx-default" attribute (unless
2211 there is a "No" in the table below) and a "xxx-supported" attribute. If the Printer object doesn't
2212 support "xxx", then it SHALL support neither an "xxx-default" attribute nor an "xxx-supported"
2213 attribute, and it SHALL treat an attribute "xxx" supplied by a client as unsupported. An attribute
2214 "xxx" may be supported for some document formats and not supported for other document
2215 formats. For example, it is expected that a Printer object would only support "orientation-
2216 requested" for some document formats (such as 'text/plain' or 'text/html') but not others (such as
2217 'application/postscript').

2218
2219 2. "xxx" is OPTIONALLY supplied by the client in a create request. If "xxx" is supplied, the client is
2220 indicating a desired job processing behavior for this Job. When "xxx" is not supplied, the client is
2221 indicating that the Printer object apply its default job processing behavior at job processing time if
2222 the document content does not contain an embedded instruction indicating an xxx-related
2223 behavior.

2224
2225 Note: Since an administrator MAY change the default value attribute after a Job object has been
2226 submitted but before it has been processed, the default value used by the Printer object at job
2227 processing time may be different that the default value in effect at job submission time.

2228
2229 3. The "xxx-supported" attribute is a Printer object attribute that describes which job processing
2230 behaviors are supported by that Printer object. A client can query the Printer object to find out
2231 what xxx-related behaviors are supported by inspecting the returned values of the "xxx-
2232 supported" attribute.

2233
2234 Note: The "xxx" in each "xxx-supported" attribute name is singular, even though an "xxx-
2235 supported" attribute usually has more than one value, such as "job-sheet-supported", unless the
2236 "xxx" Job Template attribute is plural, such as "finishings" or "sides". In such cases the "xxx-
2237 supported" attribute names are: "finishings-supported" and "sides-supported".

2238
2239 4. The "xxx-default" default value attribute describes what will be done at job processing time when
2240 no other job processing information is supplied by the client (either explicitly as an IPP attribute in
2241 the create request or implicitly as an embedded instruction within the document data).

2242
2243 If an application wishes to present an end user with a list of supported values from which to choose, the
2244 application SHOULD query the Printer object for its supported value attributes. The application
2245 SHOULD also query the default value attributes. If the application then limits selectable values to only
2246 those value that are supported, the application can guarantee that the values supplied by the client in the
2247 create request all fall within the set of supported values at the Printer. When querying the Printer, the

2248 client MAY enumerate each attribute by name in the Get-Printer-Attributes Request, or the client MAY
2249 just name the "job-template" group in order to get the complete set of supported attributes (both
2250 supported and default attributes).

2251 The "finishings" attribute is an example of a Job Template attribute. It can take on a set of values such as
2252 'staple', 'punch', and/or 'cover'. A client can query the Printer object for the "finishings-supported"
2253 attribute and the "finishings-default" attribute. The supported attribute contains a set of supported
2254 values. The default value attribute contains the finishing value(s) that will be used for a new Job if the
2255 client does not supply a "finishings" attribute in the create request and the document data does not
2256 contain any corresponding finishing instructions. If the client does supply the "finishings" attribute in the
2257 create request, the IPP object validates the value or values to make sure that they are a subset of the
2258 supported values identified in the Printer object's "finishings-supported" attribute. See section 3.2.1.2.

2259 The table below summarizes the names and relationships for all Job Template attributes. The first column
2260 of the table (labeled "Job Attribute") shows the name and syntax for each Job Template attribute in the
2261 Job object. These are the attributes that can optionally be supplied by the client in a create request. The
2262 last two columns (labeled "Printer: Default Value Attribute" and "Printer: Supported Values Attribute")
2263 shows the name and syntax for each Job Template attribute in the Printer object (the default value
2264 attribute and the supported values attribute). A "No" in the table means the Printer SHALL NOT
2265 support the attribute (that is, the attribute is simply not applicable). For brevity in the table, the 'text' and
2266 'name' entries do not show the maximum length, as in "(127)". .

2267	+	=====+	=====+	=====+
2268		Job Attribute	Printer: Default Value	Printer: Supported
2269			Attribute	Values Attribute
2270		=====+	=====+	=====+
2271		job-priority	job-priority-default	job-priority-supported
2272		(integer 1:100)	(integer 1:100)	(integer 1:100)
2273		-----+	-----+	-----+
2274		job-hold-until	job-hold-until-	job-hold-until-
2275		(type3 keyword	default	supported
2276		name)	(type3 keyword	(1setOf
2277			name)	type3 keyword name)
2278		-----+	-----+	-----+
2279		job-sheets	job-sheets-default	job-sheets-supported
2280		(type3 keyword	(type3 keyword	(1setOf
2281		name)	name)	type3 keyword name)
2282		-----+	-----+	-----+
2283		multiple-document-	multiple-document-	multiple-document-
2284		handling	handling-default	handling-supported
2285		(type2 keyword)	(type2 keyword)	(1setOf type2 keyword)
2286		-----+	-----+	-----+
2287		copies	copies-default	copies-supported
2288		(integer (1:MAX))	(integer (1:MAX))	(rangeOfInteger
2289				(1:MAX))
2290		-----+	-----+	-----+
2291		finishings	finishings-default	finishings-supported
2292		(1setOf type2 enum)	(1setOf type2 enum)	(1setOf type2 enum)
2293		-----+	-----+	-----+
2294		page-ranges	No	page-ranges-
2295		(1setOf		supported (boolean)
2296		rangeOfInteger		
2297		(1:MAX))		
2298		-----+	-----+	-----+
2299		sides	sides-default	sides-supported
2300		(type2 keyword)	(type2 keyword)	(1setOf type2 keyword)
2301		-----+	-----+	-----+
2302		number-up	number-up-default	number-up-supported
2303		(integer (1:MAX))	(integer (1:MAX))	(1setOf integer
2304				(1:MAX)
2305				rangeOfInteger
2306				(1:MAX))
2307		-----+	-----+	-----+
2308		orientation-	orientation-requested-	orientation-requested-
2309		requested	default	supported
2310		(type2 enum)	(type2 enum)	(1setOf type2 enum)
2311		-----+	-----+	-----+

2312	media	media-default	media-supported
2313	(type3 keyword	(type3 keyword	(1setOf
2314	name)	name)	type3 keyword name)
2315			
2316			media-ready
2317			(1setOf
2318			type3 keyword name)
2319	+-----+	+-----+	+-----+
2320	printer-resolution	printer-resolution-	printer-resolution-
2321	(resolution)	default	supported
2322		(resolution)	(1setOf resolution)
2323	+-----+	+-----+	+-----+
2324	print-quality	print-quality-default	print-quality-
2325	(type2 enum)	(type2 enum)	supported
2326			(1setOf type2 enum)
2327	+-----+	+-----+	+-----+
2328			
2329			

2330 4.2.1 job-priority (integer(1:100))

2331 This attribute specifies a priority for scheduling the Job. A higher value specifies a higher priority. The
 2332 value 1 indicates the lowest possible priority. The value 100 indicates the highest possible priority.
 2333 Among those jobs that are ready to print, a Printer SHALL print all jobs with a priority value of n before
 2334 printing those with a priority value of n-1 for all n.

2335 If the Printer object supports this attribute, it SHALL always support the full range from 1 to 100. No
 2336 administrative restrictions are permitted. This way an end-user can always make full use of the entire
 2337 range with any Printer object. If privileged jobs are implemented outside IPP/1.0, they SHALL have
 2338 priorities higher than 100, rather than restricting the range available to end-users.

2339 If the client does not supply this attribute and this attribute is supported by the Printer object, the Printer
 2340 object SHALL use the value of the Printer object's "job-priority-default" at job submission time (unlike
 2341 most Job Template attributes that are used if necessary at job processing time).

2342 The syntax for the "job-priority-supported" is also integer(1:100). This single integer value indicates the
 2343 number of priority levels supported. The Printer object SHALL take the value supplied by the client and
 2344 map it to the closest integer in a sequence of n integers values that are evenly distributed over the range
 2345 from 1 to 100 using the formula:

$$2346 \quad \text{roundToNearestInt}((100x+50)/n)$$

2347 where n is the value of "job-priority-supported" and x ranges from 0 through n-1.

2348 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,
2349 the sequence of values is: 17, 50 and 83; if n = 10, the sequence of values is: 5, 15, 25, 35, 45, 55, 65,
2350 75, 85, and 95; if n = 100, the sequence of values is: 1, 2, 3, ... 100.

2351 If the value of the Printer object's "job-priority-supported" is 10 and the client supplies values in the range
2352 1 to 10, the Printer object maps them to 5, in the range 11 to 20, the Printer object maps them to 15, etc.

2353 4.2.2 job-hold-until (type3 keyword | name (MAX))

2354 This attribute specifies the named time period during which the Job SHALL become a candidate for
2355 printing.

2356 Standard values for named time periods are:

2357 'no-hold': immediately, if there are not other reasons to hold the job

2358 'day-time': during the day

2359 'evening': evening

2360 'night': night

2361 'weekend': weekend

2362 'second-shift': second-shift (after close of business)

2363 'third-shift': third-shift (after midnight)

2364

2365 An administrator SHALL associate allowable print times with a named time period (by means outside
2366 IPP/1.0). An administrator is encouraged to pick names that suggest the type of time period. An
2367 administrator MAY define additional values using the 'name' or 'keyword' attribute syntax, depending on
2368 implementation.

2369 If the value of this attribute specifies a time period that is in the future, the Printer SHALL add the 'job-
2370 hold-until-specified' value to the job's "job-state-reasons" attribute, move the job to the 'pending-held'
2371 state, and SHALL NOT schedule the job for printing until the specified time-period arrives. When the
2372 specified time period arrives, the Printer SHALL remove the 'job-hold-until-specified' value from the job's
2373 "job-state-reason" attribute and, if there are no other job state reasons that keep the job in the 'pending-
2374 held' state, the Printer SHALL consider the job as a candidate for processing by moving the job to the
2375 'pending' state.

2376 If this job attribute value is the named value 'no-hold', or the specified time period has already started, the
2377 job SHALL be a candidate for processing immediately.

2378 If the client does not supply this attribute and this attribute is supported by the Printer object, the Printer
2379 object SHALL use the value of the Printer object's "job-hold-until-default" at job submission time (unlike
2380 most Job Template attributes that are used if necessary at job processing time).

2381 4.2.3 job-sheets (type3 keyword | name(MAX))

2382 This attribute determines which job start/end sheet(s), if any, SHALL be printed with a job.

2383 Standard values are:

2384 'none': no job sheet is printed

2385 'standard': one or more site specific standard job sheets are printed, e.g. a single start sheet or both
2386 start and end sheet is printed

2387

2388 An administrator MAY define additional values using the 'name' or 'keyword' attribute syntax, depending
2389 on implementation.

2390 Note: The effect of this attribute on jobs with multiple documents MAY be affected by the "multiple-
2391 document-handling" job attribute (section 4.2.4), depending on the job sheet semantics.

2392 4.2.4 multiple-document-handling (type2 keyword)

2393 This attribute is relevant only if a job consists of two or more documents. The attribute controls finishing
2394 operations and the placement of one or more print-stream pages into impressions and onto media sheets.
2395 When the value of the "copies" attribute exceeds 1, it also controls the order in which the copies that
2396 result from processing the documents are produced. For the purposes of this explanations, if "a"
2397 represents an instance of document data, then the result of processing the data in document "a" is a
2398 sequence of media sheets represented by "a(*)".

2399 Standard values are:

2400 'single-document': If a Job object has multiple documents, say, the document data is called a and b,
2401 then the result of processing all the document data (a and then b) SHALL be treated as a single
2402 sequence of media sheets for finishing operations; that is, finishing would be performed on the
2403 concatenation of the sequences a(*),b(*). The Printer object SHALL NOT force the data in each
2404 document instance to be formatted onto a new print-stream page, nor to start a new impression
2405 on a new media sheet. If more than one copy is made, the ordering of the sets of media sheets
2406 resulting from processing the document data SHALL be a(*), b(*), a(*), b(*), ..., and the Printer
2407 object SHALL force each copy (a(*),b(*)) to start on a new media sheet.

2408 'separate-documents-uncollated-copies': If a Job object has multiple documents, say, the document
2409 data is called a and b, then the result of processing the data in each document instance SHALL be
2410 treated as a single sequence of media sheets for finishing operations; that is, the sets a(*) and b(*)
2411 would each be finished separately. The Printer object SHALL force each copy of the result of
2412 processing the data in a single document to start on a new media sheet. If more than one copy is

2413 made, the ordering of the sets of media sheets resulting from processing the document data
2414 SHALL be a(*), a(*), ..., b(*), b(*)
2415 'separate-documents-collated-copies': If a Job object has multiple documents, say, the document data
2416 is called a and b, then the result of processing the data in each document instance SHALL be
2417 treated as a single sequence of media sheets for finishing operations; that is, the sets a(*) and b(*)
2418 would each be finished separately. The Printer object SHALL force each copy of the result of
2419 processing the data in a single document to start on a new media sheet. If more than one copy is
2420 made, the ordering of the sets of media sheets resulting from processing the document data
2421 SHALL be a(*), b(*), a(*), b(*),
2422

2423 The 'single-document' value is the same as 'separate-documents-collated-copies' with respect to ordering
2424 of print-stream pages, but not media sheet generation, since 'single-document' will put the first page of
2425 the next document on the back side of a sheet if an odd number of pages have been produced so far for
2426 the job, while 'separate-documents-collated-copies' always forces the next document or document copy
2427 on to a new sheet. In addition, if the "finishings" attribute specifies 'staple', then with 'single-document',
2428 documents a and b are stapled together as a single document, but with 'separate-documents-uncollated-
2429 copies' and 'separate-documents-collated-copies', documents a and b are stapled separately.

2430 Note: None of these values provide means to produce uncollated sheets within a document, i.e., where
2431 multiple copies of sheet n are produced before sheet n+1 of the same document.

2432 The relationship of this attribute and the other attributes that control document processing is described in
2433 section 16.5.

2434 4.2.5 copies (integer(1:MAX))

2435 This attribute specifies the number of copies to be printed.

2436 On many devices the supported number of collated copies will be limited by the number of physical
2437 output bins on the device, and may be different from the number of uncollated copies which can be
2438 supported.

2439 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-
2440 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other
2441 attributes that control document processing is described in section 16.5.

2442 4.2.6 finishings (1setOf type2 enum)

2443 This attribute identifies the finishing operations that the Printer uses for each copy of each printed
2444 document in the Job. For Jobs with multiple documents, the "multiple-document-handling" attribute
2445 determines what constitutes a "copy" for purposes of finishing.

2446 Standard values are:

2447	Value	Symbolic Name and Description
2448		
2449	'3'	'none': Perform no finishing
2450	'4'	'staple': Bind the document(s) with one or more staples. The exact number and placement
2451		of the staples is site-defined.
2452	'5'	'punch': This value indicates that holes are required in the finished document. The exact
2453		number and placement of the holes is site-defined. The punch specification MAY
2454		be satisfied (in a site- and implementation-specific manner) either by
2455		drilling/punching, or by substituting pre-drilled media.
2456	'6'	'cover': This value is specified when it is desired to select a non-printed (or pre-printed)
2457		cover for the document. This does not supplant the specification of a printed cover
2458		(on cover stock medium) by the document itself.
2459	'7'	'bind': This value indicates that a binding is to be applied to the document; the type and
2460		placement of the binding is site-defined."

2461
 2462 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-
 2463 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other
 2464 attributes that control document processing is described in section 16.5.

2465 If the client supplies a value of 'none' along with any other combination of values, it is the same as if only
 2466 that other combination of values had been supplied (that is the 'none' value has no effect).

2467 4.2.7 page-ranges (1setOf rangeOfInteger (1:MAX))

2468 This attribute identifies the range(s) of print-stream pages that the Printer object uses for each copy of
 2469 each document which are to be printed. Nothing is printed for any pages identified that do not exist in
 2470 the document(s). Ranges SHALL be in ascending order, for example: 1-3, 5-7, 15-19 and SHALL NOT
 2471 overlap, so that a non-spooling Printer object can process the job in a single pass. If the ranges are not
 2472 ascending or are overlapping, the IPP object SHALL reject the request and return the 'client-error-bad-
 2473 request' status code. The attribute is associated with print-stream pages not application-numbered pages
 2474 (for example, the page numbers found in the headers and or footers for certain word processing
 2475 applications).

2476 For Jobs with multiple documents, the "multiple-document-handling" attribute determines what
 2477 constitutes a "copy" for purposes of the specified page range(s). When "multiple-document-handling" is
 2478 'single-document', the Printer object SHALL apply each supplied page range once to the concatenation of
 2479 the print-stream pages. For example, if there are 8 documents of 10 pages each, the page-range '41:60'
 2480 prints the pages in the 5th and 6th documents as a single document and none of the pages of the other
 2481 documents are printed. When "multiple-document-handling" is 'separate-document-uncollated-copies' or

2482 'separate-document-collated-copies', the Printer object SHALL apply each supplied page range repeatedly
2483 to each document copy. For the same job, the page-range '1:3, 10:10' would print the first 3 pages and
2484 the 10th page of each of the 8 documents in the Job, as 8 separate documents.

2485 In most cases, the exact pages to be printed will be generated by a device driver and this attribute would
2486 not be required. However, when printing an archived document which has already been formatted, the
2487 end user may elect to print just a subset of the pages contained in the document. In this case, if page-
2488 range = n.m is specified, the first page to be printed will be page n. All subsequent pages of the document
2489 will be printed through and including page m.

2490 "page-ranges-supported" is a boolean value indicating whether or not the printer is capable of supporting
2491 the printing of page ranges. This capability may differ from one PDL to another. There is no "page-
2492 ranges-default" attribute. If the "page-ranges" attribute is not supplied by the client, all pages of the
2493 document will be printed.

2494 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-
2495 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other
2496 attributes that control document processing is described in section 16.5.

2497 4.2.8 sides (type2 keyword)

2498 This attribute specifies how print-stream pages are to be imposed upon the sides of an instance of a
2499 selected medium, i.e., an impression.

2500 The standard values are:

2501 'one-sided': imposes each consecutive print-stream page upon the same side of consecutive media
2502 sheets.

2503 'two-sided-long-edge': imposes each consecutive pair of print-stream pages upon front and back sides
2504 of consecutive media sheets, such that the orientation of each pair of print-stream pages on the
2505 medium would be correct for the reader as if for binding on the long edge. This imposition is
2506 sometimes called 'duplex' or 'head-to-head'.

2507 'two-sided-short-edge': imposes each consecutive pair of print-stream pages upon front and back sides
2508 of consecutive media sheets, such that the orientation of each pair of print-stream pages on the
2509 medium would be correct for the reader as if for binding on the short edge. This imposition is
2510 sometimes called 'tumble' or 'head-to-toe'.

2511
2512 'two-sided-long-edge', 'two-sided-short-edge', 'tumble', and 'duplex' all work the same for portrait or
2513 landscape. However 'head-to-toe' is 'tumble' in portrait but 'duplex' in landscape. 'head-to-head' also
2514 switches between 'duplex' and 'tumble' when using portrait and landscape modes.

2515 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-
 2516 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other
 2517 attributes that control document processing is described in section 16.5.

2518 4.2.9 number-up (integer(1:MAX))

2519 This attribute specifies the number of print-stream pages to impose upon a single side of an instance of a
 2520 selected medium. For example, if the value is

2521	Value	Description
2522		
2523	'1'	The Printer SHALL place one print-stream page on a single side of an instance of the
2524		selected medium (MAY add some sort of translation, scaling, or rotation).
2525	'2'	The Printer SHALL place two print-stream pages on a single side of an instance of the
2526		selected medium (MAY add some sort of translation, scaling, or rotation).
2527	'4'	The Printer SHALL place four print-stream pages on a single side of an instance of the
2528		selected medium (MAY add some sort of translation, scaling, or rotation).

2529
 2530 This attribute primarily controls the translation, scaling and rotation of print-stream pages.

2531 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-
 2532 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other
 2533 attributes that control document processing is described in section 16.5.

2534 4.2.10 orientation-requested (type2 enum)

2535 This attribute indicates the desired orientation for printed print-stream pages; it does not describe the
 2536 orientation of the client-supplied print-stream pages.

2537 For some document formats (such as 'application/postscript'), the desired orientation of the print-stream
 2538 pages is specified within the document data. This information is generated by a device driver prior to the
 2539 submission of the print job. Other document formats (such as 'text/plain') do not include the notion of
 2540 desired orientation within the document data. In the latter case it is possible for the Printer object to bind
 2541 the desired orientation to the document data after it has been submitted. It is expected that a Printer
 2542 object would only support "orientation-requested" for some document formats (e.g., 'text/plain' or
 2543 'text/html') but not others (e.g., 'application/postscript'). This is no different than any other Job Template
 2544 attribute since section 4.2, item 1, points out that a Printer object may support or not support any Job
 2545 Template attribute based on the document format supplied by the client. However, a special mention is
 2546 made here since it is very likely that a Printer object will support "orientation-requested" for only a subset
 2547 of the supported document formats.

2548 Standard values are:

2549	Value	Symbolic Name and Description
2550		
2551	'3'	'portrait': The content will be imaged across the short edge of the medium.
2552	'4'	'landscape': The content will be imaged across the long edge of the medium. Landscape is
2553		defined to be a rotation of the print-stream page to be imaged by +90 degrees with
2554		respect to the medium (i.e. anti-clockwise) from the portrait orientation. Note:
2555		The +90 direction was chosen because simple finishing on the long edge is the
2556		same edge whether portrait or landscape
2557	'5'	'reverse-landscape': The content will be imaged across the long edge of the medium.
2558		Reverse-landscape is defined to be a rotation of the print-stream page to be imaged
2559		by -90 degrees with respect to the medium (i.e. clockwise) from the portrait
2560		orientation. Note: The 'reverse-landscape' value was added because some
2561		applications rotate landscape -90 degrees from portrait, rather than +90 degrees.
2562	'6'	'reverse-portrait': The content will be imaged across the shshort edge of the medium.
2563		Reverse-portrait is defined to be a rotation of the print-stream page to be imaged
2564		by 180 degrees with respect to the medium from the portrait orientation. Note:
2565		The 'reverse-portrait' value was added for use with the "finishings" attribute in
2566		cases where the opposite edge is desired for finishing a portrait document on
2567		simple finishing devices that have only one finishing position. Thus a 'text/plain'
2568		portrait document can be stapled "on the right" by a simple finishing device as is
2569		common use with some middle eastern languages such as Hebrew.

2571 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-
2572 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other
2573 attributes that control document processing is described in section 16.5.

2574 4.2.11 media (type3 keyword | name(MAX))

2575 This attribute identifies the medium that the Printer uses for all impressions of the Job.

2576 The values for "media" include medium-names, medium-sizes, input-trays and electronic forms so that
2577 one attribute specifies the media. If a Printer object supports a medium name as a value of this attribute,
2578 such a medium name implicitly selects an input-tray that contains the specified medium. If a Printer
2579 object supports a medium size as a value of this attribute, such a medium size implicitly selects a medium
2580 name that in turn implicitly selects an input-tray that contains the medium with the specified size. If a
2581 Printer object supports an input-tray as the value of this attribute, such an input-tray implicitly selects the
2582 medium that is in that input-tray at the time the job prints. This case includes manual-feed input-trays. If
2583 a Printer object supports an electronic form as the value of this attribute, such an electronic form

2584 implicitly selects a medium-name that in turn implicitly selects an input-tray that contains the medium
 2585 specified by the electronic form. The electronic form also implicitly selects an image that the Printer
 2586 SHALL merge with the document data as its prints each page.

2587 Standard values are (taken from ISO DPA and the Printer MIB) and are listed in section 15. An
 2588 administrator MAY define additional values using the 'name' or 'keyword' attribute syntax, depending on
 2589 implementation.

2590 There is also an additional Printer attribute named "media-ready" which differs from "media-supported" in
 2591 that legal values only include the subset of "media-supported" values that are physically loaded and ready
 2592 for printing with no operator intervention required. If an IPP object supports "media-supported", it
 2593 NEED NOT support "media-ready".

2594 The relationship of this attribute and the other attributes that control document processing is described in
 2595 section 16.5.

2596 4.2.12 printer-resolution (resolution)

2597 This attribute identifies the resolution that Printer uses for the Job.

2598 4.2.13 print-quality (type2 enum)

2599 This attribute specifies the print quality that the Printer uses for the Job.

2600 The standard values are:

2601	Value	Symbolic Name and Description
2602		
2603	'3'	'draft': lowest quality available on the printer
2604	'4'	'normal': normal or intermediate quality on the printer
2605	'5'	'high': highest quality available on the printer
2606		

2607 4.3 Job Description Attributes

2608 The attributes in this section form the attribute group called "job-description". The following table
 2609 summarizes these attributes. The third column indicates whether the attribute is a MANDATORY
 2610 attribute that MUST be supported by Printer objects. If it is not indicated as MANDATORY, then it is
 2611 OPTIONAL. The maximum size in octets for 'text' and 'name' attributes is indicated in parentheses.

	Attribute	Syntax	MANDATORY?
2612			
2613			
2614			
2615	job-uri	uri	MANDATORY
2616			
2617	job-id	integer(1:MAX)	MANDATORY
2618			
2619	job-printer-uri	uri	MANDATORY
2620			
2621	job-more-info	uri	
2622			
2623	job-name	name (MAX)	MANDATORY
2624			
2625	job-originating-user-name	name (MAX)	MANDATORY
2626			
2627	job-state	type1 enum	MANDATORY
2628			
2629	job-state-reasons	1setOf type2 keyword	
2630			
2631	job-state-message	text (MAX)	
2632			
2633	number-of-documents	integer (0:MAX)	
2634			
2635	output-device-assigned	name (127)	
2636			
2637	time-at-creation	integer (0:MAX)	
2638			
2639	time-at-processing	integer (0:MAX)	
2640			
2641	time-at-completed	integer (0:MAX)	
2642			
2643	number-of-intervening-jobs	integer (0:MAX)	
2644			
2645	job-message-from-operator	text (127)	
2646			
2647	job-k-octets	integer (0:MAX)	
2648			
2649	job-impressions	integer (0:MAX)	
2650			
2651	job-media-sheets	integer (0:MAX)	
2652			
2653	job-k-octets-processed	integer (0:MAX)	
2654			
2655	job-impressions-completed	integer (0:MAX)	
2656			

2657	job-media-sheets-completed	integer (0:MAX)	
2658	+-----+-----+-----+		
2659	attributes-charset	charset	MANDATORY
2660	+-----+-----+-----+		
2661	attributes-natural-language	naturalLanguage	MANDATORY
2662	+-----+-----+-----+		
2663			
2664			

2665 4.3.1 job-uri (uri)

2666 This MANDATORY attribute contains the URI for the job. The Printer object, on receipt of a new job,
 2667 generates a URI which identifies the new Job. The Printer object returns the value of the "job-uri"
 2668 attribute as part of the response to a create request. The precise format of a Job URI is implementation
 2669 dependent. If the Printer object supports more than one URI and there is some relationship between the
 2670 newly formed Job URI and the Printer object's URI, the Printer object uses the Printer URI supplied by
 2671 the client in the create request. For example, if the create request comes in over a secure channel, the
 2672 new Job URI MUST use the same secure channel. This can guaranteed because the Printer object is
 2673 responsible for generating the Job URI is aware of its security configuration and policy as well as the
 2674 Printer URI used in the create request.

2675 For a description of this attribute and its relationship to "job-id" and "job-printer-uri" attribute, see the
 2676 discussion in section 2.4 on "Object Identity".

2677 4.3.2 job-id (integer(1:MAX))

2678 This MANDATORY attribute contains the ID of the job. The Printer, on receipt of a new job, generates
 2679 an ID which identifies the new Job on that Printer. The Printer returns the value of the "job-id" attribute
 2680 as part of the response to a create request. The 0 value is not included to allow for compatibility with
 2681 SNMP index values which also cannot be 0.

2682 For a description of this attribute and its relationship to "job-uri" and "job-printer-uri" attribute, see the
 2683 discussion in section 2.4 on "Object Identity".

2684 4.3.3 job-printer-uri (uri)

2685 This MANDATORY attribute identifies the Printer object that created this Job object. When a Printer
 2686 object creates a Job object, it populates this attribute with the Printer object URI that was used in the
 2687 create request. This attribute permits a client to identify the Printer object that created this Job object
 2688 when only the Job object's URI is available to the client. The client queries the creating Printer object to
 2689 determine which languages, charsets, operations, are supported for this Job.

2690 For a description of this attribute and its relationship to "job-uri" and "job-id" attribute, see the discussion
2691 in section 2.4 on "Object Identity".

2692 4.3.4 job-more-info (uri)

2693 Similar to "printer-more-info", this attribute contains the URI referencing some resource with more
2694 information about this Job object, perhaps an HTML page containing information about the Job.

2695 4.3.5 job-name (name(MAX))

2696 This MANDATORY attribute is the name of the job. It is a name that is more user friendly than the "job-
2697 uri" attribute value. It does not need to be unique between Jobs. The Job's "job-name" attribute is set to
2698 the value supplied by the client in the "job-name" operation attribute in the create request (see Section
2699 3.2.1.1). If, however, the "job-name" operation attribute is not supplied by the client in the create
2700 request, the Printer object, on creation of the Job, SHALL generate a name. The printer SHOULD
2701 generate the value of the Job's "job-name" attribute from the first of the following sources that produces a
2702 value: 1) the "document-name" operation attribute of the first (or only) document, 2) the "document-
2703 URI" attribute of the first (or only) document, or 3) any other piece of Job specific and/or Document
2704 Content information.

2705 4.3.6 job-originating-user-name (name(MAX))

2706 This MANDATORY attribute contains the name of the end user that submitted the print job. The Printer
2707 object sets this attribute to the most authenticated printable name that it can obtain from the
2708 authentication service over which the IPP operation was received. Only if such is not available, does the
2709 Printer object use the value supplied by the client in the "requesting-user-name" operation attribute of the
2710 create operation (see Section 8).

2711 Note: The Printer object needs to keep an internal originating user id of some form, typically as a
2712 credential of a principal, with the Job object. Since such an internal attribute is implementation-
2713 dependent and not of interest to clients, it is not specified as a Job Description attribute. This originating
2714 user id is used for authorization checks (if any) on all subsequent operation.

2715 4.3.7 job-state (type1 enum)

2716 This MANDATORY attribute identifies the current state of the job. Even though the IPP protocol
2717 defines eight values for job states, implementations only need to support those states which are
2718 appropriate for the particular implementation. In other words, a Printer supports only those job states
2719 implemented by the output device and available to the Printer object implementation.

2720 Standard values are:

2721 Values Symbolic Name and Description

2722
2723 '3' 'pending': The job is a candidate to start processing, but is not yet processing.

2724
2725 '4' 'pending-held': The job is not a candidate for processing for any number of reasons but
2726 will return to the 'pending' state as soon as the reasons are no longer present. The
2727 job's "job-state-reason" attribute SHALL indicate why the job is no longer a
2728 candidate for processing.

2729
2730 '5' 'processing': One or more of:

2731
2732 1. the job is using, or is attempting to use, one or more purely software processes
2733 that are analyzing, creating, or interpreting a PDL, etc.,
2734 2. the job is using, or is attempting to use, one or more hardware devices that are
2735 interpreting a PDL, making marks on a medium, and/or performing finishing, such
2736 as stapling, etc.,
2737 3. the Printer object has made the job ready for printing, but the output device is
2738 not yet printing it, either because the job hasn't reached the output device or
2739 because the job is queued in the output device or some other spooler, awaiting the
2740 output device to print it.

2741
2742 When the job is in the 'processing' state, the entire job state includes the detailed
2743 status represented in the printer's "printer-state", "printer-state-reasons", and
2744 "printer-state-message" attributes.

2745 Implementations MAY, though they NEED NOT, include additional values in the
2746 job's "job-state-reasons" attribute to indicate the progress of the job, such as
2747 adding the 'job-printing' value to indicate when the output device is actually
2748 making marks on paper and/or the 'processing-to-stop-point' value to indicate that
2749 the IPP object is in the process of canceling or aborting the job. Most
2750 implementations won't bother with this nuance.

2751
2752 '6' 'processing-stopped': The job has stopped while processing for any number of reasons and
2753 will return to the 'processing' state as soon as the reasons are no longer present.

2754
2755 The job's "job-state-reason" attribute MAY indicate why the job has stopped
2756 processing. For example, if the output device is stopped, the 'printer-stopped'
2757 value MAY be included in the job's "job-state-reasons" attribute.

2758

2759

2760

2761

2762

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

2763

2764

'7'

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

2765

2766

2767

2768

2769

2770

2771

2772

2773

'8'

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

2774

2775

2776

2777

2778

2779

2780

2781

2782

'9'

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

2783

2784

2785

2786

2787

2788

2789

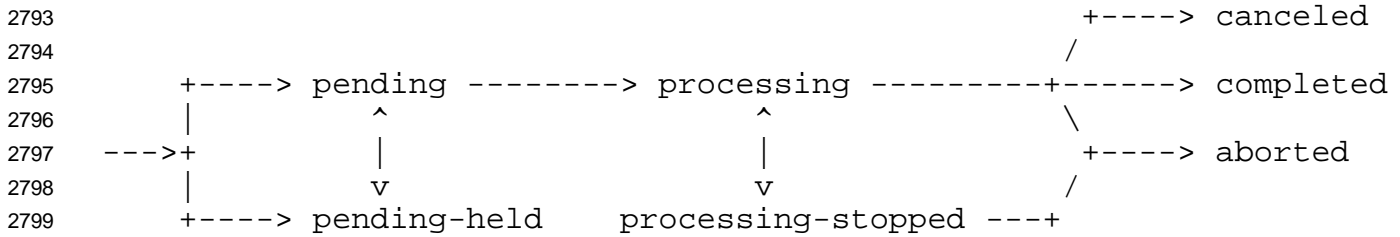
2790

2791

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

2792

The following figure shows the normal job state transitions.



2801 Normally a job progresses from left to right. Other state transitions are unlikely, but are not forbidden.
 2802 Not shown are the transitions to the 'canceled' state from the 'pending', 'pending-held', and 'processing-
 2803 stopped' states.

2804 Jobs reach one of the three terminal states: 'completed', 'canceled', or 'aborted', after the jobs have
 2805 completed all activity, including stacking output media, after the jobs have completed all activity, and all
 2806 job status attributes have reached their final values for the job.

2807 Note: As with all other IPP attributes, if the implementation can not determine the correct value for this
 2808 attribute, it may choose to respond with the out-of-band value 'unknown' rather than try to guess at some
 2809 possibly incorrect value and give the end user the wrong impression about the state of the Job object.
 2810 For example, if the implementation is just a gateway into some printing system that does not provide
 2811 detailed status about the print job, the IPP Job object's state might literally be 'unknown'.

2812 4.3.8 job-state-reasons (1setOf type2 keyword)

2813 This attribute provides additional information about the job's current state, i.e., information that augments
 2814 the value of the job's "job-state" attribute.

2815 Implementation of these values is OPTIONAL, i.e., a Printer NEED NOT implement them, even if (1)
 2816 the output device supports the functionality represented by the reason and (2) is available to the Printer
 2817 object implementation. These values MAY be used with any job state or states for which the reason
 2818 makes sense. Furthermore, when implemented, the Printer SHALL return these values when the reason
 2819 applies and SHALL NOT return them when the reason no longer applies whether the value of the Job's
 2820 "job-state" attribute changed or not. When the Job does not have any reasons for being in its current
 2821 state, the value of the Job's "job-state-reasons" attribute SHALL be 'none'.

2822 Note: While values cannot be added to the 'job-state' attribute without impacting deployed clients that
 2823 take actions upon receiving "job-state" values, it is the intent that additional "job-state-reasons" values
 2824 can be defined and registered without impacting such deployed clients. In other words, the "job-state-
 2825 reasons" attribute is intended to be extensible.

2826 The following standard values are defined. For ease of understanding, the values are presented in the
 2827 order in which the reasons are likely to occur (if implemented), starting with the 'job-incoming' value:

2828 'none': There are no reasons for the job's current state.
2829 'job-incoming': The CreateJob operation has been accepted by the Printer, but the Printer is expecting
2830 additional Send-Document and/or Send-URI operations and/or is accessing/accepting document
2831 data.
2832 'submission-interrupted': The job was not completely submitted for some unforeseen reason, such as:
2833 (1) the Printer has crashed before the job was closed by the client, (2) the Printer or the document
2834 transfer method has crashed in some non-recoverable way before the document data was entirely
2835 transferred to the Printer, (3) the client crashed or failed to close the job before the time-out
2836 period.
2837 'job-outgoing': The Printer is transmitting the job to the output device.
2838 'job-hold-until-specified': The value of the job's "job-hold-until" attribute was specified with a time
2839 period that is still in the future. The job SHALL NOT be a candidate for processing until this
2840 reason is removed and there are no other reasons to hold the job.
2841 'resources-are-not-ready': At least one of the resources needed by the job, such as media, fonts,
2842 resource objects, etc., is not ready on any of the physical printer's for which the job is a candidate.
2843 This condition MAY be detected when the job is accepted, or subsequently while the job is
2844 pending or processing, depending on implementation. The job may remain in its current state or
2845 be moved to the 'pending-held' state, depending on implementation and/or job scheduling policy.
2846 'printer-stopped-partly': The value of the Printer's "printer-state-reasons" attribute contains the value
2847 'stopped-partly'.
2848 'printer-stopped': The value of the Printer's "printer-state" attribute is 'stopped'.
2849 'job-interpreting': Job is in the 'processing' state, but more specifically, the Printer is interpreting the
2850 document data.
2851 'job-queued': Job is in the 'processing' state, but more specifically, the Printer has queued the
2852 document data.
2853 'job-transforming': Job is in the 'processing' state, but more specifically, the Printer is interpreting
2854 document data and producing another electronic representation.
2855 'job-printing': The output device is marking media. This value is useful for Printers which spend a
2856 great deal of time processing (1) when no marking is happening and then want to show that
2857 marking is now happening or (2) when the job is in the process of being canceled or aborted while
2858 the job remains in the 'processing' state, but the marking has not yet stopped so that impression or
2859 sheet counts are still increasing for the job.
2860 'job-canceled-by-user': The job was canceled by the owner of the job using the Cancel-Job request,
2861 i.e., by a user whose authenticated identity is the same as the value of the originating user that
2862 created the Job object, or by some other authorized end-user, such as a member of the job owner's
2863 security group.
2864 'job-canceled-by-operator': The job was canceled by the operator using the Cancel-Job request, i.e.,
2865 by a user who has been authenticated as having operator privileges (whether local or remote). If
2866 the security policy is to allow anyone to cancel anyone's job, then this value may be used when the

2867 job is canceled by other than the owner of the job. For such a security policy, in effect, everyone
2868 is an operator as far as canceling jobs with IPP is concerned.

2869 'job-canceled-at-device': The job was canceled by an unidentified local user, i.e., a user at a console
2870 at the device.

2871 'aborted-by-system': The job (1) is in the process of being aborted, (2) has been aborted by the
2872 system and placed in the 'aborted' state, or (3) has been aborted by the system and placed in the
2873 'pending-held' state, so that a user or operator can manually try the job again.

2874 'processing-to-stop-point': The requester has issued a Cancel-job operation or the Printer object has
2875 aborted the job, but is still performing some actions on the job until a specified stop point occurs
2876 or job termination/cleanup is completed.

2877

2878 This reason is recommended to be used in conjunction with the 'processing' job state to indicate
2879 that the Printer object is still performing some actions on the job while the job remains in the
2880 'processing' state. After all the job's job description attributes have stopped incrementing, the
2881 Printer object moves the job from the 'processing' state to the 'canceled' or 'aborted' job states.

2882

2883 'service-off-line': The Printer is off-line and accepting no jobs. All 'pending' jobs are put into the
2884 'pending-held' state. This situation could be true if the service's or document transform's input is
2885 impaired or broken.

2886 'job-completed-successfully': The job completed successfully.

2887 'job-completed-with-warnings': The job completed with warnings.

2888 'job-completed-with-errors': The job completed with errors (and possibly warnings too).

2889

2890 4.3.9 job-state-message (text(MAX))

2891 This attribute specifies information about the "job-state" and "job-state-reasons" attributes in human
2892 readable text. If the Printer object supports this attribute, the Printer object SHALL be able to generate
2893 this message in any of the natural languages identified by the Printer's "generated-natural-language-
2894 supported" attribute (see the "attributes-natural-language" operation attribute specified in Section
2895 3.1.4.1).

2896 Note: the value SHOULD NOT contain additional information not contained in the values of the "job-
2897 state" and "job-states-reasons" attributes, such as interpreter error information. Otherwise, application
2898 programs might attempt to parse the (localized text). For such additional information such as interpreter
2899 errors for application program consumption, a new attribute with keyword values, needs to be developed
2900 and registered.

2901 4.3.10 number-of-documents (integer(0:MAX))

2902 This attribute indicates the number of documents in the job, i.e., the number of Send-Document, Send-
2903 URI, Print-Job, or Print-URI operations that the Printer has accepted for this job, regardless of whether
2904 the document data has reached the Printer object or not.

2905 Implementations supporting the OPTIONAL Create-Job/Send-Document/Send-URI operations
2906 SHOULD support this attribute so that clients can query the number of documents in each job.

2907 4.3.11 output-device-assigned (name(127))

2908 This attribute identifies the output device to which the Printer object has assigned this job. If an output
2909 device implements an embedded Printer object, the Printer object NEED NOT set this attribute. If a print
2910 server implements a Printer object, the value MAY be empty (zero-length string) or not returned until the
2911 Printer object assigns an output device to the job. This attribute is particularly useful when a single
2912 Printer object support multiple devices (so called "fan-out").

2913 4.3.12 time-at-creation (integer(0:MAX))

2914 This attribute indicates the point in time at which the Job object was created. In order to populate this
2915 attribute, the Printer object uses the value in its "printer-up-time" attribute at the time the Job object is
2916 created.

2917 4.3.13 time-at-processing (integer(0:MAX))

2918 This attribute indicates the point in time at which the Job object began processing. In order to populate
2919 this attribute, the Printer object uses the value in its "printer-up-time" attribute at the time the Job object
2920 is moved into the 'processing' state for the first time.

2921 4.3.14 time-at-completed (integer(0:MAX))

2922 This attribute indicates the point in time at which the Job object completed (or was cancelled or aborted).
2923 In order to populate this attribute, the Printer object uses the value in its "printer-up-time" attribute at the
2924 time the Job object is moved into the 'completed' or 'canceled' or 'aborted' state.

2925 4.3.15 number-of-intervening-jobs (integer(0:MAX))

2926 This attribute indicates the number of jobs that are "ahead" of this job in the relative chronological order
2927 of expected time to complete (i.e., the current scheduled order). For efficiency, it is only necessary to
2928 calculate this value when an operation is performed that requests this attribute.

2929 4.3.16 job-message-from-operator (text(127))

2930 This attribute provides a message from an operator, system administrator or "intelligent" process to
2931 indicate to the end user the reasons for modification or other management action taken on a job.

2932 4.3.17 job-k-octets (integer(0:MAX))

2933 This attribute specifies the total size of the document(s) in K octets, i.e., in units of 1024 octets requested
2934 to be processed in the job. The value SHALL be rounded up, so that a job between 1 and 1024 octets
2935 SHALL be indicated as being 1, 1025 to 2048 SHALL be 2, etc.

2936 This value SHALL NOT include the multiplicative factors contributed by the number of copies specified
2937 by the "copies" attribute, independent of whether the device can process multiple copies without making
2938 multiple passes over the job or document data and independent of whether the output is collated or not.
2939 Thus the value is independent of the implementation and indicates the size of the document(s) measured
2940 in K octets independent of the number of copies.

2941 This value SHALL also not include the multiplicative factor due to a copies instruction embedded in the
2942 document data. If the document data actually includes replications of the document data, this value will
2943 include such replication. In other words, this value is always the size of the source document data, rather
2944 than a measure of the hardcopy output to be produced.

2945 Note: This attribute and the following two attributes ("job-impressions" and "job-media-sheets") are not
2946 intended to be counters; they are intended to be useful routing and scheduling information if known. For
2947 these three attributes, the Printer object may try to compute the value if it is not supplied in the create
2948 request. Even if the client does supply a value for these three attributes in the create request, the Printer
2949 object MAY choose to change the value if the Printer object is able to compute a value which is more
2950 accurate than the client supplied value. The Printer object may be able to determine the correct value for
2951 these three attributes either right at job submission time or at any later point in time.

2952 4.3.18 job-impressions (integer(0:MAX))

2953 This attribute specifies the total size in number of impressions of the document(s) being submitted (see
2954 the definition of impression in section 13.2.5).

2955 As with "job-k-octets", this value SHALL NOT include the multiplicative factors contributed by the
2956 number of copies specified by the "copies" attribute, independent of whether the device can process
2957 multiple copies without making multiple passes over the job or document data and independent of
2958 whether the output is collated or not. Thus the value is independent of the implementation and reflects
2959 the size of the document(s) measured in impressions independent of the number of copies.

2960 As with "job-k-octets", this value SHALL also not include the multiplicative factor due to a copies
2961 instruction embedded in the document data. If the document data actually includes replications of the
2962 document data, this value will include such replication. In other words, this value is always the number of
2963 impressions in the source document data, rather than a measure of the number of impressions to be
2964 produced by the job.

2965 See the Note in the "job-k-octets" attribute that also applies to this attribute.

2966 4.3.19 job-media-sheets (integer(0:MAX))

2967 This attribute specifies the total number of media sheets to be produced for this job.

2968 Unlike the "job-k-octets" and the "job-impressions" attributes, this value SHALL include the
2969 multiplicative factors contributed by the number of copies specified by the "copies" attribute and a
2970 'number of copies' instruction embedded in the document data, if any. This difference allows the system
2971 administrator to control the lower and upper bounds of both (1) the size of the document(s) with "job-k-
2972 octets-supported" and "job-impressions-supported" and (2) the size of the job with "job-media-sheets-
2973 supported".

2974 See the Note in the "job-k-octets" attribute that also applies to this attribute.

2975 4.3.20 job-k-octets-processed (integer(0:MAX))

2976 This attribute specifies the total number of octets processed in K octets, i.e., in units of 1024 octets so
2977 far. The value SHALL be rounded up, so that a job between 1 and 1024 octets inclusive SHALL be
2978 indicated as being 1, 1025 to 2048 inclusive SHALL be 2, etc.

2979 For implementations where multiple copies are produced by the interpreter with only a single pass over
2980 the data, the final value SHALL be equal to the value of the "job-k-octets" attribute. For
2981 implementations where multiple copies are produced by the interpreter by processing the data for each
2982 copy, the final value SHALL be a multiple of the value of the "job-k-octets" attribute.

2983 Note: This attribute and the following two attributes ("job-impressions-completed" and "job-sheets-
2984 completed") are intended to be counters. That is, the value for a job that has not started processing
2985 SHALL be 0. When the job's "job-state" is 'processing' or 'processing-stopped', this value is intended to
2986 contain the amount of the job that has been processed to the time at which the attributes are requested.

2987 4.3.21 job-impressions-completed (integer(0:MAX))

2988 This job attribute specifies the number of impressions completed for the job so far. For printing devices,
2989 the impressions completed includes interpreting, marking, and stacking the output.

2990 See the note in "job-k-octets-processed" which also applies to this attribute.

2991 4.3.22 job-media-sheets-completed (integer(0:MAX))

2992 This job attribute specifies the media-sheets completed marking and stacking for the entire job so far
2993 whether those sheets have been processed on one side or on both.

2994 See the note in "job-k-octets-processed" which also applies to this attribute.

2995 4.3.23 attributes-charset (charset)

2996 This MANDATORY attribute is populated using the value in the client supplied "attributes-charset"
2997 attribute in the create request. It identifies the charset (coded character set and encoding method) used
2998 by any Job attributes with attribute syntax 'text' and 'name' that were supplied by the client in the create
2999 request. See Section 3.1.4 for a complete description of the "attributes-charset" operation attribute.

3000 This attribute does not indicate the charset in which the 'text' and 'name' values are stored internally in the
3001 Job object. The internal charset is implementation-defined. The IPP object SHALL convert from
3002 whatever the internal charset is to that being requested in an operation as specified in Section 3.1.4.

3003 4.3.24 attributes-natural-language (naturalLanguage)

3004 This MANDATORY attribute is populated using the value in the client supplied "attributes-natural-
3005 language" attribute in the create request. It identifies the natural language used for any Job attributes
3006 with attribute syntax 'text' and 'name' that were supplied by the client in the create request. See Section
3007 3.1.4 for a complete description of the "attributes-natural-language" operation attribute. See Section
3008 3.2.6 for how this attribute is returned in a Get-Jobs operation when jobs with different natural languages
3009 are returned. See Sections 4.1.1.2 and 4.1.2.2 for how a Natural Language Override may be supplied
3010 explicitly for each 'text' and 'name' attribute value that differs from the value identified by the "attributes-
3011 natural-language" attribute.

3012 4.4 Printer Description Attributes

3013 These attributes form the attribute group called "printer-description". The following table summarizes
3014 these attributes, their syntax, and whether or not they are MANDATORY for a Printer object to support.
3015 If they are not indicated as MANDATORY, they are OPTIONAL. The maximum size in octets for 'text'
3016 and 'name' attributes is indicated in parentheses.

3017 Note: How these attributes are set by an Administrator is outside the scope of this specification.

3018	+-----+-----+-----+
3019	Attribute Syntax MANDATORY?
3020	+-----+-----+-----+
3021	printer-uri-supported 1setOf uri MANDATORY
3022	+-----+-----+-----+
3023	uri-security-supported 1setOf type2 keyword MANDATORY
3024	+-----+-----+-----+
3025	printer-name name (127) MANDATORY
3026	+-----+-----+-----+
3027	printer-location text (127)
3028	+-----+-----+-----+
3029	printer-info text (127)
3030	+-----+-----+-----+
3031	printer-more-info uri
3032	+-----+-----+-----+
3033	printer-driver-installer uri
3034	+-----+-----+-----+
3035	printer-make-and-model text (127)
3036	+-----+-----+-----+
3037	printer-more-info-
3038	manufacturer uri
3039	+-----+-----+-----+
3040	printer-state type1 enum MANDATORY
3041	+-----+-----+-----+
3042	printer-state-reasons 1setOf type2 keyword
3043	+-----+-----+-----+
3044	printer-state-message text (MAX)
3045	+-----+-----+-----+
3046	operations-supported 1setOf type2 enum MANDATORY
3047	+-----+-----+-----+
3048	charset-configured charset MANDATORY
3049	+-----+-----+-----+
3050	charset-supported 1setOf charset MANDATORY
3051	+-----+-----+-----+
3052	natural-language-configured naturalLanguage MANDATORY
3053	+-----+-----+-----+
3054	generated-natural-language-
3055	supported 1setOf MANDATORY
3056	naturalLanguage
3057	+-----+-----+-----+
3058	document-format-default mimeType MANDATORY
3059	+-----+-----+-----+
3060	document-format-
3061	supported 1setOf MANDATORY
3062	mimeType
3062	printer-is-accepting-jobs boolean MANDATORY

3063	+-----+-----+-----+
3064	queued-job-count integer (0:MAX)
3065	+-----+-----+-----+
3066	printer-message-from- text (127)
3067	operator
3068	+-----+-----+-----+
3069	color-supported boolean
3070	+-----+-----+-----+
3071	reference-uri-schemes- lsetOf uriScheme
3072	supported
3073	+-----+-----+-----+
3074	pdl-override-supported type2 keyword MANDATORY
3075	+-----+-----+-----+
3076	printer-up-time integer (1:MAX) MANDATORY
3077	+-----+-----+-----+
3078	printer-current-time dateTime
3079	+-----+-----+-----+
3080	multiple-operation-time-out integer (1:MAX)
3081	+-----+-----+-----+
3082	compression-supported lsetOf type3 keyword
3083	+-----+-----+-----+
3084	job-k-octets-supported rangeOfInteger
3085	(0:MAX)
3086	+-----+-----+-----+
3087	job-impressions-supported rangeOfInteger
3088	(0:MAX)
3089	+-----+-----+-----+
3090	job-media-sheets-supported rangeOfInteger
3091	(0:MAX)
3092	+-----+-----+-----+
3093	

3094 4.4.1 printer-uri-supported (lsetOf uri)

3095 This MANDATORY Printer attribute contains at least one URI for the Printer object. It OPTIONALLY
 3096 contains more than one URI for the Printer object. An administrator determines a Printer object's
 3097 URI(s) and configures this attribute to contain those URIs by some means outside the scope of IPP/1.0.
 3098 The precise format of this URI is implementation dependent and depends on the protocol. See the next
 3099 section for a description "uri-security-supported" which is the MANDATORY companion attribute to
 3100 this "printer-uri-supported" attribute. See section 2.4 on Printer object identity and section 8.2 on
 3101 security and URIs for more information.

3102 4.4.2 uri-security-supported (1setOf type2 keyword)

3103 This MANDATORY Printer attribute MUST have the same cardinality (contain the same number of
3104 values) as the "printer-uri-supported" attribute. This attribute identifies the security mechanisms used for
3105 each URI listed in the "printer-uri-supported" attribute. The "i th" value in "uri-security-supported"
3106 corresponds to the "i th" value in "printer-uri-supported" and it describes the security mechanisms used
3107 for accessing the Printer object via that URI. The following standard values are defined:

3108 'none': There are no secure communication channel protocols in use for the given URI.

3109 'tls': TLS 1.0 [TLS] is the secure communications channel protocol in use for the given URI.

3110 'ssl3': SSL3 is the secure communications channel protocol in use for the given URI.

3111

3112 Consider the following example. For a single Printer object, an administrator configures the "printer-uri-
3113 supported" and "uri-security-supported" attributes as follows:

3114 "printer-uri-supported": 'http://acme.com/open-use-printer', 'http://acme.com/restricted-use-printer',

3115 'http://acme.com/private-printer'

3116 "uri-security-supported": 'none', 'none', 'tls'

3117

3118 In this case, one Printer object has three URIs.

- 3119 - For the first URI, 'http://acme.com/open-use-printer', the value 'none' in "uri-security-supported"
3120 indicates that there is no secure channel protocol configured to run under HTTP. The name
3121 implies that there is no Basic or Digest authentication being used, but it is up to the client to
3122 determine that while using HTTP underneath the IPP application protocol.
- 3123 - For the second URI, 'http://acme.com/restricted-use-printer', the value 'none' in "uri-security-
3124 supported" indicates that there is no secure channel protocol configured to run under HTTP. In
3125 this case, although the name does imply that there is some sort of Basic or Digest authentication
3126 being used within HTTP, it is up to the client to determine that while using HTTP and by
3127 processing any '401 Unauthorized' HTTP error messages.
- 3128 - For the third URI, 'http://acme.com/private-printer', the value 'tls' in "uri-security-supported"
3129 indicates that TLS is being used to secure the channel. The client SHOULD be prepared to use
3130 TLS framing to negotiate an acceptable ciphersuite to use while communicating with the Printer
3131 object. In this case, the name implies the use of a secure communications channel, but the fact is
3132 made explicit by the presence of the 'tls' value in "uri-security-supported". The client does not
3133 need to resort to understanding which security it must use by following naming conventions or by
3134 parsing the URI to determine which security mechanisms are implied.

3135

3136 It is expected that many IPP Printer objects will be configured to support only one channel (either
3137 configured to use TLS access or not), and will therefore only ever have one URI listed in the "printer-uri-

3138 supported" attribute. No matter the configuration of the Printer object (whether it has only one URI or
3139 more than one URI), a client SHALL supply only one URI in the target "printer-uri" operation attribute.

3140 4.4.3 printer-name (name(127))

3141 This MANDATORY Printer attribute contains the name of the Printer object. It is a name that is more
3142 end-user friendly than a URI. An administrator determines a printer's name and sets this attribute to that
3143 name. This name may be the last part of the printer's URI or it may be unrelated. In non-US-English
3144 locales, a name may contain characters that are not allowed in a URI.

3145 4.4.4 printer-location (text(127))

3146 This Printer attribute identifies the location of the device. This could include things like: "in Room 123A,
3147 second floor of building XYZ".

3148 4.4.5 printer-info (text(127))

3149 This Printer attribute identifies the descriptive information about this Printer object. This could include
3150 things like: "This printer can be used for printing color transparencies for HR presentations", or "Out of
3151 courtesy for others, please print only small (1-5 page) jobs at this printer", or even "This printer is going
3152 away on July 1, 1997, please find a new printer".

3153 4.4.6 printer-more-info (uri)

3154 This Printer attribute contains a URI used to obtain more information about this specific Printer object.
3155 For example, this could be an HTTP type URI referencing an HTML page accessible to a Web Browser.
3156 The information obtained from this URI is intended for end user consumption. Features outside the scope
3157 of IPP can be accessed from this URI. The information is intended to be specific to this printer instance
3158 and site specific services (e.g. job pricing, services offered, end user assistance). The device manufacturer
3159 may initially populate this attribute.

3160 4.4.7 printer-driver-installer (uri)

3161 This Printer attribute contains a URI to use to locate the driver installer for this Printer object. This
3162 attribute is intended for consumption by automata. The mechanics of print driver installation is outside
3163 the scope of IPP. The device manufacturer may initially populate this attribute.

3164 4.4.8 printer-make-and-model (text(127))

3165 This Printer attribute identifies the make and model of the device. The device manufacturer may initially
3166 populate this attribute.

3167 4.4.9 printer-more-info-manufacturer (uri)

3168 This Printer attribute contains a URI used to obtain more information about this type of device. The
3169 information obtained from this URI is intended for end user consumption. Features outside the scope of
3170 IPP can be accessed from this URI (e.g., latest firmware, upgrades, print drivers, optional features
3171 available, details on color support). The information is intended to be germane to this printer without
3172 regard to site specific modifications or services. The device manufacturer may initially populate this
3173 attribute.

3174 4.4.10 printer-state (type1 enum)

3175 This MANDATORY Printer attribute identifies the current state of the device. The "printer-state
3176 reasons" attribute augments the "printer-state" attribute to give more detailed information about the
3177 Printer in the given printer state.

3178 A Printer object need only update this attribute before responding to an operation which requests the
3179 attribute; the Printer object NEED NOT update this attribute continually, since asynchronous event
3180 notification is not part of IPP/1.0. A Printer NEED NOT implement all values if they are not applicable
3181 to a given implementation.

3182 The following standard values are defined:

3183	Value	Symbolic Name and Description
3184		
3185	'3'	'idle': If a Printer receives a job (whose required resources are ready) while in this state,
3186		such a job SHALL transit into the processing state immediately. If the printer-
3187		state-reasons attribute contains any reasons, they SHALL be reasons that would
3188		not prevent a job from transiting into the processing state immediately, e.g., toner-
3189		low. Note: if a Printer controls more than one output device, the above definition
3190		implies that a Printer is idle if at least one output device is idle.
3191		
3192	'4'	'processing': If a Printer receives a job (whose required resources are ready) while in this
3193		state, such a job SHALL transit into the pending state immediately. Such a job
3194		SHALL transit into the processing state only after jobs ahead of it complete. If the
3195		printer-state-reasons attribute contains any reasons, they SHALL be reasons that
3196		do not prevent the current job from printing, e.g. toner-low. Note: if a Printer

3197 controls more than one output device, the above definition implies that a Printer is
3198 processing if at least one output device is processing, and none is idle.

3199
3200 '5' 'stopped': If a Printer receives a job (whose required resources are ready) while in this
3201 state, such a job SHALL transit into the pending state immediately. Such a job
3202 SHALL transit into the processing state only after some human fixes the problem
3203 that stopped the printer and after jobs ahead of it complete printing. If supported,
3204 the "printer-state-reasons" attribute SHALL contain at least one reason, e.g.
3205 media-jam, which prevents it from either processing the current job or transitioning
3206 a pending job to the processing state.

3207
3208 Note: if a Printer controls more than one output device, the above definition
3209 implies that a Printer is stopped only if all output devices are stopped. Also, it is
3210 tempting to define stopped as when a sufficient number of output devices are
3211 stopped and leave it to an implementation to define the sufficient number. But
3212 such a rule complicates the definition of stopped and processing. For example,
3213 with this alternate definition of stopped, a job can move from idle to processing
3214 without human intervention, even though the Printer is stopped.

3215

3216 4.4.11 printer-state-reasons (1setOf type2 keyword)

3217 This Printer attribute supplies additional detail about the device's state.

3218 Each keyword value MAY have a suffix to indicate its level of severity. The three levels are: report (least
3219 severe), warning, and error (most severe).

3220 - '-report': This suffix indicates that the reason is a "report". An implementation may choose to omit
3221 some or all reports. Some reports specify finer granularity about the printer state; others serve as
3222 a precursor to a warning. A report SHALL contain nothing that could affect the printed output.

3223 - '-warning': This suffix indicates that the reason is a "warning". An implementation may choose to
3224 omit some or all warnings. Warnings serve as a precursor to an error. A warning SHALL contain
3225 nothing that prevents a job from completing, though in some cases the output may be of lower
3226 quality.

3227 - '-error': This suffix indicates that the reason is an "error". An implementation SHALL include all
3228 errors. If this attribute contains one or more errors, printer SHALL be in the stopped state.

3229

3230 If the implementation does not add any one of the three suffixes, all parties SHALL assume that the
3231 reason is an "error".

3232 If a Printer object controls more than one output device, each value of this attribute MAY apply to one or
3233 more of the output devices. An error on one output device that does not stop the Printer object as a
3234 whole MAY appear as a warning in the Printer's "printer-state-reasons" attribute. If the "printer-state"
3235 for such a Printer has a value of 'stopped', then there MUST be an error reason among the values in the
3236 "printer-state-reasons" attribute.

3237 The following standard values are defined:

3238 'other': The device has detected an error other than one listed in this document.

3239 'none': There are not reasons. This state reason is semantically equivalent to "printer-state-reasons"
3240 without any value.

3241 'media-needed': A tray has run out of media.

3242 'media-jam': The device has a media jam.

3243 'paused': Someone has paused the Printer object. In this state, a Printer SHALL NOT produce
3244 printed output, but it SHALL perform other operations requested by a client. If a Printer had
3245 been printing a job when the Printer was paused, the Printer SHALL resume printing that job
3246 when the Printer is no longer paused and leave no evidence in the printed output of such a pause.

3247 'shutdown': Someone has removed a Printer object from service, and the device may be powered
3248 down or physically removed. In this state, a Printer object SHALL NOT produce printed output,
3249 and unless the Printer object is realized by a print server that is still active, the Printer object
3250 SHALL perform no other operations requested by a client, including returning this value. If a
3251 Printer object had been printing a job when it was shutdown, the Printer need not resume printing
3252 that job when the Printer is no longer shutdown. If the Printer resumes printing such a job, it may
3253 leave evidence in the printed output of such a shutdown, e.g. the part printed before the shutdown
3254 may be printed a second time after the shutdown.

3255 'connecting-to-device': The Printer object has scheduled a job on the output device and is in the
3256 process of connecting to a shared network output device (and might not be able to actually start
3257 printing the job for an arbitrarily long time depending on the usage of the output device by other
3258 servers on the network).

3259 'timed-out': The server was able to connect to the output device (or is always connected), but was
3260 unable to get a response from the output device.

3261 'stopping': The Printer object is in the process of stopping the device and will be stopped in a while.
3262 When the device is stopped, the Printer object will change the Printer object's state to 'stopped'.
3263 The 'stopping-warning' reason is never an error, even for a Printer with a single output device.
3264 When an output-device ceases accepting jobs, the Printer will have this reason while the output
3265 device completes printing.

3266 'stopped-partly': When a Printer object controls more than one output device, this reason indicates
3267 that one or more output devices are stopped. If the reason is a report, fewer than half of the
3268 output devices are stopped. If the reason is a warning, fewer than all of the output devices are
3269 stopped.

3270 'toner-low': The device is low on toner.

3271 'marker-supply-low': The device is low on marker supply (ink, paint, etc.).

3272 'spool-area-full': The limit of persistent storage allocated for spooling has been reached.

3273 'cover-open': One or more covers on the device are open.

3274 'interlock-open': One or more interlock devices on the printer are unlocked.

3275 'door-open': One or more doors on the device are open.

3276 'input-tray-missing': One or more input trays are not in the device.

3277 'media-low': At least one input tray is low on media.

3278 'media-empty': At least one input tray is empty.

3279 'output-tray-missing': One or more output trays are not in the device

3280 'output-area-almost-full': One or more output area is almost full (e.g. tray, stacker, collator).

3281 'output-area-full': One or more output area is full. (e.g. tray, stacker, collator)

3282 'marker-supply-low': The device is low on at least one marker supply. (e.g. toner, ink, ribbon)

3283 'marker-supply-empty': The device is out of at least one marker supply. (e.g. toner, ink, ribbon)

3284 'marker-waste-almost-full': The device marker supply waste receptacle is almost full.

3285 'marker-waste-full': The device marker supply waste receptacle is full.

3286 'fuser-over-temp': The fuser temperature is above normal.

3287 'fuser-under-temp': The fuser temperature is below normal.

3288 'opc-near-eol': The optical photo conductor is near end of life.

3289 'opc-life-over': The optical photo conductor is no longer functioning.

3290 'developer-low': The device is low on developer.

3291 'developer-empty': The device is out of developer.

3292 'interpreter-resource-unavailable': An interpreter resource is unavailable (i.e. font, form)

3293

3294 4.4.12 printer-state-message (text(MAX))

3295 This Printer attribute specifies the additional information about the printer state and printer state reasons
3296 in human readable text. If the Printer object supports this attribute, the Printer object SHALL be able to
3297 generate this message in any of the natural languages identified by the Printer's "generated-natural-
3298 language-supported" attribute (see the "attributes-natural-language" operation attribute specified in
3299 Section 3.1.4.1).

3300 4.4.13 operations-supported (1setOf type2 enum)

3301 This MANDATORY Printer attribute specifies the set of supported operations for this Printer object and
 3302 contained Job objects. No 32-bit enum value for this attribute SHALL exceed 0x8FFF, since these values
 3303 are passed in two octets in each Protocol request [IPP-PRO].

3304 The following standard values are defined:

3305	Value	Operation Name
3306	-----	-----
3307		
3308	0x0000	reserved, not used
3309	0x0001	reserved, not used
3310	0x0002	Print-Job
3311	0x0003	Print-URI
3312	0x0004	Validate-Job
3313	0x0005	Create-Job
3314	0x0006	Send-Document
3315	0x0007	Send-URI
3316	0x0008	Cancel-Job
3317	0x0009	Get-Job-Attributes
3318	0x000A	Get-Jobs
3319	0x000B	Get-Printer-Attributes
3320	0x000C-0x3FFF	reserved for future operations
3321	0x4000-0x8FFF	reserved for private extensions

3322

3323 This allows for certain vendors to implement private extensions that are guaranteed to not conflict with
 3324 future registered extensions. However, there is no guarantee that two or more private extensions will not
 3325 conflict.

3326 4.4.14 charset-configured (charset)

3327 This MANDATORY Printer attribute identifies the charset that the Printer object has been configured to
 3328 represent 'text' and 'name' Printer attributes that are set by the operator, system administrator, or
 3329 manufacturer, i.e., for "printer-name" (name), "printer-location" (text), "printer-info" (text), and "printer-
 3330 make-and-model" (text). Therefore, the value of the Printer object's "charset-configured" attribute
 3331 SHALL also be among the values of the Printer object's "charset-supported" attribute.

3332 4.4.15 charset-supported (1setOf charset)

3333 This MANDATORY Printer attribute identifies the set of charsets that the Printer and contained Job
3334 objects support in attributes with attribute syntax 'text' and 'name'. At least the value 'utf-8' SHALL be
3335 present, since IPP objects MUST support the UTF-8 [RFC2044] charset. If a Printer object supports a
3336 charset, it means that for all attributes of syntax 'text' and 'name' the IPP object SHALL (1) accept the
3337 charset in requests and return the charset in responses as needed.

3338 If more charsets than UTF-8 are supported, the IPP object SHALL perform charset conversion between
3339 the charsets as described in Section 3.2.1.2.

3340 4.4.16 natural-language-configured (naturalLanguage)

3341 This MANDATORY Printer attribute identifies the natural language that the Printer object has been
3342 configured to represent 'text' and 'name' Printer attributes that are set by the operator, system
3343 administrator, or manufacturer, i.e., for "printer-name" (name), "printer-location" (text), "printer-info"
3344 (text), and "printer-make-and-model" (text). When returning these Printer attributes, the Printer object
3345 MAY return them in the configured natural language specified by this attribute, instead of the natural
3346 language requested by the client in the "attributes-natural-language" operation attribute. See Section
3347 3.1.4.1 for the specification of the OPTIONAL multiple natural language support. Therefore, the value
3348 of the Printer object's "natural-language-configured" attribute SHALL also be among the values of the
3349 Printer object's "generated-natural-language-supported" attribute.

3350 4.4.17 generated-natural-language-supported (1setOf naturalLanguage)

3351 This MANDATORY Printer attribute identifies the natural language(s) that the Printer object and
3352 contained Job objects support in attributes with attribute syntax 'text' and 'name'. The natural language(s)
3353 supported depends on implementation and/or configuration. Unlike charsets, IPP objects SHALL accept
3354 requests with any natural language or any Natural Language Override whether the natural language is
3355 supported or not.

3356 If a Printer object supports a natural language, it means that for any of the attributes for which the Printer
3357 or Job object generates messages, i.e., for the "job-state-message" and "printer-state-message" attributes
3358 and Operation Messages (see Section 3.1.5) in operation responses, the Printer and Job objects SHALL
3359 be able to generate messages in any of the Printer's supported natural languages. See section 3.1.4 for
3360 the specification of 'text' and 'name' attributes in operation requests and responses.

3361 Note: A Printer object that supports multiple natural languages, often has separate catalogs of messages,
3362 one for each natural language supported.

3363 4.4.18 document-format-default (mimeMediaType)

3364 This Printer attribute identifies the document format that the Printer object has been configured to assume
3365 if the client does not supply a "document-format" operation attribute in any of the operation requests that
3366 supply document data. The standard values for this attribute are Internet Media types (sometimes called
3367 MIME types). For further details see the description of the 'mimeMediaType' attribute syntax in Section
3368 4.1.9.

3369 4.4.19 document-format-supported (1setOf mimeMediaType)

3370 This Printer attribute identifies the set of document formats that the Printer object and contained Job
3371 objects can support. For further details see the description of the 'mimeMediaType' attribute syntax in
3372 Section 4.1.9.

3373 4.4.20 printer-is-accepting-jobs (boolean)

3374 This MANDATORY Printer attribute indicates whether the printer is currently able to accept jobs, i.e., is
3375 accepting Print-Job, Print-URI, and Create-Job requests. If the value is 'true', the printer is accepting
3376 jobs. If the value is 'false', the Printer object is currently rejecting any jobs submitted to it. In this case,
3377 the Printer object returns the 'server-error-not-accepting-jobs' status code.

3378 Note: This value is independent of the "printer-state" and "printer-state-reasons" attributes because its
3379 value does not affect the current job; rather it affects future jobs. This attribute may cause the Printer to
3380 reject jobs when the "printer-state" is 'idle' or it may cause the Printer object to accept jobs when the
3381 "printer-state" is 'stopped'.

3382 4.4.21 queued-job-count (integer(0:MAX))

3383 This Printer attribute contains a count of the number of jobs that are either 'pending', 'processing',
3384 'pending-held', or 'processing-stopped' and is set by the Printer object.

3385 4.4.22 printer-message-from-operator (text(127))

3386 This Printer attribute provides a message from an operator, system administrator or "intelligent" process
3387 to indicate to the end user information or status of the printer, such as why it is unavailable or when it is
3388 expected to be available.

3389 4.4.23 color-supported (boolean)

3390 This Printer attribute identifies whether the device is capable of any type of color printing at all, including
3391 highlight color. All document instructions having to do with color are embedded within the document
3392 PDL (none are external IPP attributes in IPP/1.0).

3393 Note: end-users are able to determine the nature and details of the color support by querying the
3394 "printer-more-info-manufacturer" Printer attribute.

3395 4.4.24 reference-uri-schemes-supported (1setOf uriScheme)

3396 This Printer attribute specifies which URI schemes are supported for use in the "document-uri" operation
3397 attribute of the Print-URI or Send-URI operation. If a Printer object supports these optional operations,
3398 it **MUST** support the "reference-uri-schemes-supported" Printer attribute with at least the following
3399 schemed URI values:

3400 'ftp': The Printer object will use an FTP 'get' operation as defined in RFC 1738 and RFC 2316.

3401
3402 The Printer object **MAY OPTIONALLY** support other URI schemes (see section 4.1.6).

3403 4.4.25 pdl-override-supported (type2 keyword)

3404 This **MANDATORY** Printer attribute expresses the ability for a particular Printer implementation to
3405 either attempt to override document data instructions with IPP attributes or not.

3406 This attribute takes on the following values:

- 3407 - 'attempted': This value indicates that the Printer object attempts to make the IPP attribute values
3408 take precedence over embedded instructions in the document data, however there is no guarantee.
- 3409 - 'not-attempted': This value indicates that the Printer object makes no attempt to make the IPP
3410 attribute values take precedence over embedded instructions in the document data.

3411
3412 Section 16 contains a full description of how this attribute interacts with and affects other IPP attributes,
3413 especially the "ipp-attribute-fidelity" attribute.

3414 4.4.26 printer-up-time (integer(1:MAX))

3415 This **MANDATORY** Printer attribute indicates the amount of time (in seconds) that this instance of this
3416 Printer implementation has been up and running. This value is used to populate the Job attributes "time-
3417 at-creation", "time-at-processing", and "time-at-completed". These time values are all measured in
3418 seconds and all have meaning only relative to this attribute, "printer-up-time". The value is a

3419 monotonically increasing value starting from 1 when the Printer object is started-up (initialized, booted,
3420 etc.).

3421 If the Printer object goes down at some value 'n', and comes back up, the implementation MAY:

- 3422 1. Know how long it has been down, and resume at some value greater than 'n', or
- 3423 2. Restart from 1.

3424

3425 In the first case, the Printer SHOULD not tweak any existing related Job attributes ("time-at-creation",
3426 "time-at-processing", and "time-at-completed"). In the second case, the Printer object SHOULD reset
3427 those attributes to 0. If a client queries a time-related Job attribute and finds the value to be 0, the client
3428 MUST assume that the Job was submitted in some life other than the Printer's current life.

3429 4.4.27 printer-current-time (dateTime)

3430 This Printer attribute indicates the current absolute wall-clock time. If an implementation supports this
3431 attribute, then a client could calculate the absolute wall-clock time each Job's "time-at-creation", "time-at-
3432 processing", and "time-at-completed" attributes by using both "printer-up-time" and this attribute,
3433 "printer-current-time". If an implementation does not support this attribute, a client can only calculate
3434 the relative time of certain events based on the MANDATORY "printer-up-time" attribute.

3435 4.4.28 multiple-operation-time-out (integer(1:MAX))

3436 This Printer attributes identifies how long (in seconds) the Printer object waits for additional Send-
3437 Document or Send-URI operations to follow a still-open multi-document Job object before taking one of
3438 the actions indicated in section 3.3.1.

3439 4.4.29 compression-supported (1setOf type3 keyword)

3440 This Printer attribute identifies the set of supported compression algorithms for document data.
3441 Compression only applies to the document data; compression does not apply to the encoding of the IPP
3442 operation itself. The supported values are used to validate the client supplied "compression" operation
3443 attributes in Print-Job, Send-Document, and Send-URI requests.

3444 Standard values are :

- 3445 'none': no compression is used.
- 3446 'deflate': ZIP public domain inflate/deflate) compression technology
- 3447 'gzip' GNU zip compression technology described in RFC 1952 [RFC1952].
- 3448 'compress': UNIX compression technology

3449

3450 4.4.30 job-k-octets-supported (rangeOfInteger(0:MAX))

3451 This Printer attribute specifies the upper and lower bounds of total sizes of jobs in K octets, i.e., in units
3452 of 1024 octets. The supported values are used to validate the client supplied "job-k-octets" operation
3453 attributes in create requests. The corresponding job description attribute "job-k-octets" is defined in
3454 section 4.3.17.

3455 4.4.31 job-impressions-supported (rangeOfInteger(0:MAX))

3456 This Printer attribute specifies the upper and lower bounds for the number of impressions per job. The
3457 supported values are used to validate the client supplied "job-impressions" operation attributes in create
3458 requests. The corresponding job description attribute "job-impressions" is defined in section 4.3.18.

3459 4.4.32 job-media-sheets-supported (rangeOfInteger(0:MAX))

3460 This Printer attribute specifies the upper and lower bounds for the number of media sheets per job. The
3461 supported values are used to validate the client supplied "job-media-sheets" operation attributes in create
3462 requests. The corresponding Job attribute "job-media-sheets" is defined in section 4.3.19.

3463 5. Conformance

3464 This section describes conformance issues and requirements. This document introduces model entities
3465 such as objects, operations, attributes, attribute syntaxes, and attribute values. These conformance
3466 sections describe the conformance requirements which apply to these model entities.

3467 5.1 Client Conformance Requirements

3468 A conforming client SHALL support all MANDATORY operations as defined in this document. For
3469 each attribute included in an operation request, a conforming client SHALL supply a value whose type
3470 and value syntax conforms to the requirements of the Model document as specified in Sections 3 and 4.
3471 A conforming client MAY supply any registered extensions and/or private extensions in an operation
3472 request, as long as they meet the requirements in Section 6.

3473 Otherwise, there are no conformance requirements placed on the user interfaces provided by IPP clients
3474 or their applications. For example, one application might not allow an end user to submit multiple
3475 documents per job, while another does. One application might first query a Printer object in order to

3476 supply a graphical user interface (GUI) dialogue box with supported and default values whereas a
3477 different implementation might not.

3478 When sending a request, an IPP client **NEED NOT** supply any attributes that are indicated as
3479 **OPTIONALLY** supplied by the client.

3480 A client **SHALL** be able to accept any of the attribute syntaxes defined in Section 4.1, including their full
3481 range, that may be returned to it in a response from a Printer object. For presentation purposes,
3482 truncation of long attribute values is not recommended. A recommended approach would be for the
3483 client implementation to allow the user to scroll through long attribute values.

3484 A query response may contain attribute groups, attributes, and values that the client does not expect.
3485 Therefore, a client implementation **MUST** gracefully handle such responses and not refuse to inter-
3486 operate with a conforming Printer that is returning extended registered or private attributes and/or
3487 attribute values that conform to Section 6. Clients may choose to ignore any parameters, attributes, or
3488 values that they do not understand.

3489 5.2 IPP Object Conformance Requirements

3490 This section specifies the conformance requirements for conforming implementations with respect to
3491 objects, operations, and attributes.

3492 5.2.1 Objects

3493 Conforming implementations **SHALL** implement all of the model objects as defined in this specification in
3494 the indicated sections:

3495 Section 2.1 - Printer Object

3496 Section 2.2 - Job Object

3497

3498 5.2.2 Operations

3499 Conforming IPP object implementations **SHALL** implement all of the **MANDATORY** model operations,
3500 including mandatory responses, as defined in this specification in the indicated sections:

3501 For a Printer object:

3502	Print-Job (section 3.2.1)	MANDATORY
3503	Print-URI (section 3.2.2)	OPTIONAL
3504	Validate-Job (section 3.2.3)	MANDATORY
3505	Create-Job (section 3.2.4)	OPTIONAL

3506	Get-Printer-Attributes (section 3.2.5)	MANDATORY
3507	Get-Jobs (section 3.2.6)	MANDATORY
3508		
3509	For a Job object:	
3510	Send-Document (section 3.3.1)	OPTIONAL
3511	Send-URI (section 3.3.2)	OPTIONAL
3512	Cancel-Job (section 3.3.3)	MANDATORY
3513	Get-Job-Attributes (section 3.3.4)	MANDATORY
3514		

3515 Conforming IPP objects SHALL support all MANDATORY operation attributes and all values of such
3516 attributes if so indicated in the description. Conforming IPP objects SHALL ignore all unsupported or
3517 unknown operation attributes or operation attribute groups received in a request, but SHALL reject a
3518 request that contains a supported operation attribute that contains an unsupported value.

3519 The following section on object attributes specifies the support required for object attributes.

3520 5.2.3 IPP Object Attributes

3521 Conforming IPP objects SHALL support all of the MANDATORY object attributes, as defined in this
3522 specification in the indicated sections.

3523 If an object supports an attribute, it SHALL support only those values specified in this document or
3524 through the extension mechanism described in section 5.2.4. It MAY support any non-empty subset of
3525 these values. That is, it SHALL support at least one of the specified values and at most all of them.

3526 5.2.4 Extensions

3527 A conforming IPP object MAY support registered extensions and private extensions, as long as they meet
3528 the requirements specified in Section 6.

3529 For each attribute included in an operation response, a conforming IPP object SHALL return a value
3530 whose type and value syntax conforms to the requirement of the Model document as specified in Sections
3531 3 and 4.

3532 5.2.5 Attribute Syntaxes

3533 An IPP object SHALL be able to accept any of the attribute syntaxes defined in Section 4.1, including
3534 their full range, in any operation in which a client may supply attributes or the system administrator may
3535 configure attributes (by means outside the scope of IPP/1.0). Furthermore, an IPP object SHALL return

3536 attributes to the client in operation responses that conform to the syntax specified in Section 4.1,
3537 including their full range if supplied previously by a client.

3538 5.3 Charset and Natural Language Requirements

3539 All clients and IPP objects SHALL support the 'utf-8' charset as defined in section 4.1.7.

3540 IPP objects MUST be able to accept any client request which correctly uses the "attributes-natural-
3541 language" operation attribute or the Natural Language Override mechanism on any individual attribute
3542 whether or not the natural language is supported by the IPP object. If an IPP object supports a natural
3543 language, then it MUST be able to translate (perhaps by table lookup) all generated 'text' or 'name'
3544 attribute values into one of the supported languages (see section 3.1.4). That is, the IPP object that
3545 supports a natural language NEED NOT be a general purpose translator of any arbitrary 'text' or 'name'
3546 value supplied by the client into that natural language. However, the object MUST be able to translate
3547 (automatically generate) any of its own attribute values and messages into that natural language.

3548 5.4 Security Conformance Requirements

3549 Conforming IPP Printer objects MAY support Transport Layer Security (TLS) access, support access
3550 without TLS or support both means of access.

3551 Conforming IPP clients SHOULD support TLS access and non-TLS access. Note: This client
3552 requirement to support both means that conforming IPP clients will be able to inter-operate with any IPP
3553 Printer object.

3554 For a detailed discussion of security considerations and the IPP application security profile required for
3555 TLS support, see section 8.

3556 6. IANA Considerations (registered and private extensions)

3557 This section describes how IPP can be extended to allow the following registered and private extensions
3558 to IPP:

- 3559 1. keyword attribute values
- 3560 2. enum attribute values
- 3561 3. attributes
- 3562 4. attribute syntaxes
- 3563 5. operations

3564 6. status codes

3565
3566 Registered and private extensions registered for use with IPP/1.0 are OPTIONAL for client and IPP
3567 object conformance to the IPP/1.0 Model specification.

3568 These extension procedures are aligned with the guidelines as set forth by the IESG [IANA-CON].
3569 Section 12 describes how to propose new registrations for consideration. IANA will reject registration
3570 proposals that leave out required information or do not follow the appropriate format described in
3571 Section 12.

3572 6.1 Typed 'keyword' and 'enum' Extensions

3573 IPP allows for 'keyword' and 'enum' extensions (see sections 4.1.3 and 4.1.4). This document uses
3574 prefixes to the 'keyword' and 'enum' basic attribute syntax type in order to communicate extra information
3575 to the reader through its name. This extra information need not be represented in the protocol because it
3576 is unimportant to a client or Printer object. The list below describes the prefixes and their meaning.

3577 "type1": The IPP specification must be revised to add a new keyword or a new enum. No private
3578 keywords or enums are allowed.

3579
3580 "type2": Implementers can, at any time, add new keyword or enum values by proposing the complete
3581 specification to IANA:

3582
3583 iana@iana.org

3584
3585 IANA will forward the registration proposal to the IPP Designated Expert who will review the
3586 proposal with a mailing list that the Designated Expert keeps for this purpose. Initially, that list
3587 will be the mailing list used by the IPP WG:

3588
3589 ipp@pwg.org

3590
3591 even after the IPP WG is disbanded as permitted by [IANA-CON]. The IPP Designated Expert is
3592 appointed by the IESG Area Director responsible for IPP, according to [IANA-CON].

3593
3594 When a type2 keyword or enum is approved, the IPP Designated Expert becomes the point of
3595 contact for any future maintenance that might be required for that registration.

3596
3597 "type3": Implementers can, at any time, add new keyword and enum values by submitting the
3598 complete specification to IANA as for type2 who will forward the proposal to the IPP Designated
3599 Expert. While no additional technical review is required, the IPP Designated Expert may, at

3600 his/her discretion, forward the proposal to the same mailing list as for type2 registrations for
3601 advice and comment.

3602
3603 When a type3 keyword or enum is approved by the IPP Designated Expert, the original proposer
3604 becomes the point of contact for any future maintenance that might be required for that
3605 registration.

3606
3607 For type2 and type3 keywords, the proposer includes the name of the keyword in the registration
3608 proposal and the name is part of the technical review.

3609 After type2 and type3 enums specifications are approved, the IPP Designated Expert in consultation with
3610 IANA assigns the next available enum number for each enum value.

3611 IANA will publish approved type2 and type3 keyword and enum attributes value registration
3612 specifications in:

3613 <ftp://isi.edu/iana/assignments/ipp/attribute-values/xxx/yyy.txt>

3614 where xxx is the attribute name that specifies the initial values and yyy.txt is a descriptive file name that
3615 contains one or more enums or keywords approved at the same time. For example, if several additional
3616 enums for stapling are approved for use with the "finishings" attribute (and "finishings-default" and
3617 "finishings-supported" attributes), IANA will publish the additional values in the file:

3618 <ftp://isi.edu/iana/assignments/ipp/attribute-values/finishings/stapling.txt>.

3619 Note: Some attributes are defined to be either 'type3 keywords' and 'name' which allows for attribute
3620 values to be extended by a site administrator with administrator defined names. Such names are not
3621 registered with IANA.

3622 By definition, each of the three types above assert some sort of registry or review process in order for
3623 extensions to be considered valid. Each higher numbered level (1, 2, 3) tends to be decreasingly less
3624 stringent than the previous level. Therefore, any typeN value MAY be registered using a process for
3625 some typeM where M is less than N, however such registration is NOT REQUIRED. For example, a
3626 type3 value MAY be registered in a type 1 manner (by being included in a future version of an IPP
3627 specification), however, it is NOT REQUIRED.

3628 This specification defines keyword and enum values for all of the above types, including type3 keywords.

3629 For private (unregistered) keyword extensions, implementers SHOULD use keywords with a suitable
3630 distinguishing prefix, such as "xxx-" where xxx is the (lowercase) fully qualified company name registered
3631 with IANA for use in domain names [RFC1035]. For example, if the company XYZ Corp. had obtained
3632 the domain name "XYZ.com", then a private keyword 'abc' would be: 'xyz.com-abc'.

3633 Note: RFC 1035 [RFC1035] indicates that while upper and lower case letters are allowed in domain
3634 names, no significance is attached to the case. That is, two names with the same spelling but different
3635 case are to be treated as if identical. Also, the labels in a domain name must follow the rules for
3636 ARPANET host names: They must start with a letter, end with a letter or digit, and have as interior
3637 characters only letters, digits, and hyphen. Labels must be 63 characters or less. Labels are separated by
3638 the "." character.

3639 For private (unregistered) enum extension, implementers SHALL use values in the reserved integer range
3640 which is 2^{30} to $2^{31}-1$.

3641 6.2 Attribute Extensibility

3642 Attribute names are type2 keywords. Therefore, new attributes may be registered and have the same
3643 status as attributes in this document by following the type2 extension rules. For private (unregistered)
3644 attribute extensions, implementers SHOULD use keywords with a suitable distinguishing prefix as
3645 described in Section 6.1.

3646 IANA will publish approved attribute registration specifications as separate files:

3647 <ftp://isi.edu/iana/assignments/ipp/attributes/xxx-yyy.txt>

3648 where "xxx-yyy" is the new attribute name.

3649 If a new Printer object attribute is defined and its values can be affected by a specific document format, its
3650 specification needs to contain the following sentence:

3651 "The value of this attribute returned in a Get-Printer-Attributes response MAY depend on the
3652 "document-format" attribute supplied (see Section 3.2.5.1)."

3653 If the specification does not, then its value in the Get-Printer-Attributes response SHALL NOT depend
3654 on the "document-format" supplied in the request. When a new Job Template attribute is registered, the
3655 value of the Printer attributes MAY vary with "document-format" supplied in the request without the
3656 specification having to indicate so.

3657 6.3 Attribute Syntax Extensibility

3658 Attribute syntaxes are like type2 enums. Therefore, new attribute syntaxes may be registered and have
3659 the same status as attribute syntaxes in this document by following the type2 extension rules described in
3660 Section 6.1. The value codes that identify each of the attribute syntaxes are assigned in the protocol
3661 specification [IPP-PRO].

3662 For attribute syntaxes, the IPP Designated Expert in consultation with IANA assigns the next attribute
3663 syntax code in the appropriate range as specified in [IPP-PRO]. IANA will publish approved attribute
3664 syntax registration specifications as separate files:

3665 <ftp.isi.edu/iana/assignments/ipp/attribute-syntaxes/xxx-yyy.txt>

3666 where 'xxx-yyy' is the new attribute syntax name.

3667 6.4 Operation Extensibility

3668 Operations may also be registered following the type2 procedures described in Section 6.1, though major
3669 new operations will usually be done by a new standards track RFC that augments this document. For
3670 private (unregistered) operation extensions, implementers SHALL use the range for the "operation-id" in
3671 requests specified in Section 4.4.13 "operations-supported" Printer attribute.

3672 For operations, the IPP Designated Expert in consultation with IANA assigns the next operation-id code
3673 as specified in Section 4.4.13. IANA will publish approved operation registration specifications as
3674 separate files:

3675 <ftp.isi.edu/iana/assignments/ipp/operations/Xxx-Yyy.txt>

3676 where "Xxx-Yyy" is the new operation name.

3677 6.5 Status Code Extensibility

3678 Operation status codes may also be registered following the type2 procedures described in Section 6.1.
3679 The values for status codes are allocated in ranges as specified in Section 14 for each status code class:

3680 "informational" - Request received, continuing process
3681 "successful" - The action was successfully received, understood, and accepted
3682 "redirection" - Further action must be taken in order to complete the request
3683 "client-error" - The request contains bad syntax or cannot be fulfilled
3684 "server-error" - The IPP object failed to fulfill an apparently valid request

3685
3686 For private (unregistered) operation status code extensions, implementers SHALL use the top of each
3687 range as specified in Section 14.

3688 For operation status codes, the IPP Designated Expert in consultation with IANA assigns the next status
3689 code in the appropriate class range as specified in Section 14. IANA will publish approved status code
3690 registration specifications as separate files:

3691 <ftp.isi.edu/iana/assignments/ipp/status-codes/xxx-yyy.txt>

3692 where "xxx-yyy" is the new operation status code keyword.

3693 6.6 Registration of MIME types/sub-types for document-formats

3694 The "document-format" attribute's syntax is 'mimeMediaType'. This means that valid values are Internet
3695 Media Types (see Section 4.1.9). RFC 2045 [RFC2045] defines the syntax for valid Internet media
3696 types. IANA is the registry for all Internet media types.

3697 6.7 Registration of charsets for use in 'charset' attribute values

3698 The "attributes-charset" attribute's syntax is 'charset'. This means that valid values are charsets names.
3699 When a charset in the IANA registry has more than one name (alias), the name labeled as "(preferred
3700 MIME name)", if present, SHALL be used (see Section 4.1.7). IANA is the registry for charsets
3701 following the procedures of [IANA-CSa].

3702 7. Internationalization Considerations

3703 Some of the attributes have values that are text strings and names which are intended for human
3704 understanding rather than machine understanding (see the 'text' and 'name' attribute syntaxes in Sections
3705 4.1.1 and 4.1.2).

3706 In each operation request, the client

- 3707 - identifies the charset and natural language of the request which affects each supplied 'text' and 'name'
3708 attribute value, and
- 3709 - requests the charset and natural language for attributes returned by the IPP object in operation
3710 responses (as described in Section 3.1.4.1).

3711
3712 In addition, the client MAY separately and individually identify the Natural Language Override of a
3713 supplied 'text' or 'name' attribute using the 'textWithLanguage' and 'nameWithLanguage' technique
3714 described section 4.1.1.2 and 4.1.2.2 respectively.

3715 All IPP objects SHALL support the UTF-8 [RFC2044] charset in all 'text' and 'name' attributes
3716 supported. If an IPP object supports more than the UTF-8 charset, the object SHALL convert between
3717 them in order to return the requested charset to the client according to Section 3.1.4.2. If an IPP object
3718 supports more than one natural language, the object SHOULD return 'text' and 'name' values in the
3719 natural language requested where those values are generated by the Printer (see Section 3.1.4.1).

3720 For Printers that support multiple charsets and/or multiple natural languages in 'text' and 'name' attributes,
3721 different jobs may have been submitted in differing charsets and/or natural languages. All responses
3722 SHALL be returned in the charset requested by the client. However, the Get-Jobs operation uses the
3723 'textWithLanguage' and 'nameWithLanguage' mechanism to identify the differing natural languages with
3724 each job returned.

3725 The Printer object also has configured charset and natural language attributes. The client can query the
3726 Printer object to determine the list of charsets and natural languages supported by the Printer object and
3727 what the Printer object's configured values are. See the "charset-configured", "charset-supported",
3728 "natural-language-configured", and "generated-natural-language-supported" Printer description attributes
3729 for more details.

3730 The "charset-supported" attributed identifies the supported charsets. If a charset is supported, the IPP
3731 object MUST be capable of converting to and from that charset into any other supported charset. In
3732 many cases, an IPP object will support only one charset and it MUST be the UTF-8 charset.

3733 The "charset-configured" attribute identifies the one supported charset which is the native charset given
3734 the current configuration of the IPP object (administrator defined).

3735 The "generated-natural-language-supported" attribute identifies the set of supported natural languages for
3736 generated messages; it is not related to the set of natural languages that must be accepted for client
3737 supplied 'text' and 'name' attributes. For client supplied 'text' and 'name' attributes, an IPP object MUST
3738 accept ALL supplied natural languages. Just because a Printer object is currently configured to support
3739 'en-us' natural language does not mean that the Printer object should reject a job if the client supplies a
3740 job name that is in 'fr-ca'.

3741 The "natural-language-configured" attribute identifies the one supported natural language for generated
3742 messages which is the native natural language given the current configuration of the IPP object
3743 (administrator defined).

3744 Attributes of type 'text' and 'name' are populated from different sources. These attributes can be
3745 categorized into following groups (depending on the source of the attribute):

- 3746 1. Some attributes are supplied by the client (e.g., the client supplied "job-name", "document-name",
3747 and "requesting-user-name" operation attributes along with the corresponding Job object's "job-
3748 name" and "job-originating-user-name" attributes). The IPP object MUST accept these attributes
3749 in any natural language no matter what the set of supported languages for generated messages
- 3750 2. Some attributes are supplied by the system administrator (e.g., the Printer object's "printer-name"
3751 and "printer-location" attributes). These too can be in any natural language. If the natural
3752 language for these attributes is different than what a client requests, then they must be reported
3753 using the Natural Language Override mechanism.

- 3754 3. Some attributes are supplied by the device manufacturer (e.g., the Printer object's "printer-make-
 3755 and-model" attribute). These too can be in any natural language. If the natural language for these
 3756 attributes is different than what a client requests, then they must be reported using the Natural
 3757 Language Override mechanism.
- 3758 4. Some attributes are supplied by the operator (e.g., the Job object's "job-message-from-operator"
 3759 attribute). These too can be in any natural language. If the natural language for these attributes is
 3760 different than what a client requests, then they must be reported using the Natural Language
 3761 Override mechanism.
- 3762 5. Some attributes are generated by the IPP object (e.g., the Job object's "job-state-message"
 3763 attribute, the Printer object's "printer-state-message" attribute, and the "status-message" operation
 3764 attribute). These attributes can only be in one of the "generated-natural-language-supported"
 3765 natural languages. If a client requests some natural language for these attributes other than one of
 3766 the supported values, the IPP object SHOULD respond in using the value of the "natural-
 3767 language-configured" attribute (using the Natural Language Override mechanism if needed).
 3768

3769 The 'text' and 'name' attributes specified in this version of this document (additional ones will be
 3770 registered according to the procedures in Section 6) are:

3771	Attributes	Source
3772	-----	-----
3773	Operation Attributes	
3774	job-name (name)	client
3775	document-name (name)	client
3776	requesting-user-name (name)	client
3777		
3778	Job Attributes:	
3779	job-name (name)	client or Printer object
3780	job-originating-user-name (name)	Printer object
3781	job-state-message (text)	Job or Printer object
3782	job-message-from-operator (text)	operator
3783		
3784	Printer Attributes:	
3785	printer-name (name)	administrator
3786	printer-location (text)	administrator
3787	printer-info (text)	administrator
3788	printer-make-and-model (text)	administrator or manufacturer
3789	printer-state-message (text)	Printer object
3790	printer-message-from-operator (text)	operator
3791		

3792 8. Security Considerations

3793 Some IPP objects MAY be deployed over protocol stacks that support Transport Layer Security (TLS)
3794 Version 1.0. Other IPP objects MAY be deployed over protocol stacks that do not support TLS. Some
3795 IPP objects MAY be deployed over both types of protocol stacks. Those IPP objects that support TLS,
3796 are capable of supporting mutual authentication as well as privacy of messages via multiple encryption
3797 schemes. TLS 1.0 also supports a backwards compatibility mode for negotiating down to SSL3 which
3798 leverages the vast installed base of SSL3 aware clients and servers. An important point about security
3799 related information for TLS access to an IPP object, is that the security-related parameters
3800 (authentication, encryption keys, etc.) are "out-of-band" to the actual IPP protocol.

3801 An IPP object that does not support TLS MAY elect to support a transport layer that provides other
3802 security mechanisms. For example, in a mapping of IPP over HTTP/1.1 [IPP-PRO], if the IPP object
3803 does not support TLS, HTTP still allows for client authentication.

3804 It is difficult to anticipate the security risks that might exist in any given IPP environment. For example, if
3805 IPP is used within a given corporation over a private network, the risks of exposing document data may
3806 be low enough that the corporation will choose not to use encryption on that data. However, if the
3807 connection between the client and the IPP object is over a public network, the client may wish to protect
3808 the content of the information during transmission through the network with encryption.

3809 Furthermore, the value of the information being printed may vary from one IPP environment to the next.
3810 Printing payroll checks, for example, would have a different value than printing public information from a
3811 file. There is also the possibility of denial-of-service attacks, but denial-of-service attacks against printing
3812 resources are not well understood and there is no published precedents regarding this scenario.

3813 Once the authenticated identity of the requester has been supplied to the IPP object, the object uses that
3814 identity to enforce any authorization policy that might be in place. For example, one site's policy might
3815 be that only the job owner is allowed to cancel a job. The details and mechanisms to set up a particular
3816 access control policy are not part of IPP/1.0, and must be established via some other type of
3817 administrative or access control framework. However, there are operation status codes that allow an IPP
3818 server to return information back to a client about any potential access control violations for an IPP
3819 object.

3820 During a create operation, the client's identity is recorded in the Job object in an implementation-defined
3821 attribute. This information can be used to verify a client's identity for subsequent operations on that Job
3822 object in order to enforce any access control policy that might be in effect. See section 8.3 below for
3823 more details.

3824 Since the security levels or the specific threats that any given IPP system administrator may be concerned
3825 with cannot be anticipated, IPP MUST be capable of operating with different security mechanisms and

3826 security policies as required by the individual installation. Security policies might vary from very strong,
3827 to very weak, to none at all, and corresponding security mechanisms will be required. TLS Version 1.0
3828 supports the type of negotiated levels of security required by most, if not all, potential IPP environments.
3829 IPP environments that require no security can elect to deploy IPP objects that do not utilize the optional
3830 TLS security mechanisms.

3831 8.1 Security Scenarios

3832 The following sections describe specific security attacks for IPP environments. Where examples are
3833 provided they should be considered illustrative of the environment and not an exhaustive set. Not all of
3834 these environments will necessarily be addressed in initial implementations of IPP.

3835 8.1.1 Client and Server in the Same Security Domain

3836 This environment is typical of internal networks where traditional office workers print the output of
3837 personal productivity applications on shared work-group printers, or where batch applications print their
3838 output on large production printers. Although the identity of the user may be trusted in this environment,
3839 a user might want to protect the content of a document against such attacks as eavesdropping, replaying
3840 or tampering.

3841 8.1.2 Client and Server in Different Security Domains

3842 Examples of this environment include printing a document created by the client on a publicly available
3843 printer, such as at a commercial print shop; or printing a document remotely on a business associate's
3844 printer. This latter operation is functionally equivalent to sending the document to the business associate
3845 as a facsimile. Printing sensitive information on a Printer in a different security domain requires strong
3846 security measures. In this environment authentication of the printer is required as well as protection
3847 against unauthorized use of print resources. Since the document crosses security domains, protection
3848 against eavesdropping and document tampering are also required. It will also be important in this
3849 environment to protect Printers against "spamming" and malicious document content.

3850 8.1.3 Print by Reference

3851 When the document is not stored on the client, printing can be done by reference. That is, the print
3852 request can contain a reference, or pointer, to the document instead of the actual document itself.
3853 Standard methods currently do not exist for remote entities to "assume" the credentials of a client for
3854 forwarding requests to a 3rd party. It is anticipated that Print-By-Reference will be used to access
3855 "public" documents and that sophisticated methods for authenticating "proxies" will not be specified for
3856 version 1 of IPP.

3857 8.2 URIs for TLS and non-TLS Access

3858 As described earlier, an IPP object can support TLS access, non-TLS access, or both. The "printer-uri-
3859 supported" attribute contains the Printer object's URI(s). Its companion attribute, "uri-security-
3860 supported", identifies the security mechanism used for each URI listed in the "printer-uri-supported"
3861 attribute. For each Printer operation request, a client SHALL supply only one URI in the "printer-uri"
3862 operation attribute. In other words, even though the Printer supports more than one URI, the client only
3863 interacts with the Printer object using one of its URIs. This duality is not needed for Job objects, since the
3864 Printer objects is the factory for Job objects, and the Printer object will generate the correct URI for new
3865 Job objects depending on the Printer object's security configuration.

3866 8.3 The "requesting-user-name" (name(MAX)) Operation Attribute

3867 Each operation SHALL specify the user who is performing the operation in both of the following two
3868 ways:

- 3869 1) via the MANDATORY "requesting-user-name" operation attribute that a client SHOULD supply
3870 in all operations. The client SHALL obtain the value for this attribute from an environmental or
3871 network login name for the user, rather than allowing the user to supply any value. If the client
3872 does not supply a value for "requesting-user-name", the printer SHALL assume that the client is
3873 supplying some anonymous name, such as "anonymous".
- 3874 2) via an authentication mechanism of the underlying transport which may be configured to give no
3875 authentication information.

3876
3877 There are six cases to consider:

- 3878 a) the authentication mechanism gives no information, and the client doesn't specify "requesting-
3879 user-name".
- 3880 b) the authentication mechanism gives no information, but the client specifies "requesting-user-
3881 name".
- 3882 c) the authentication mechanism specifies a user which has no human readable representation, and the
3883 client doesn't specify "requesting-user-name".
- 3884 d) the authentication mechanism specifies a user which has no human readable representation, but the
3885 client specifies "requesting-user-name".
- 3886 e) the authentication mechanism specifies a user which has a human readable representation. The
3887 Printer object ignores the "requesting-user-name".
- 3888 f) the authentication mechanism specifies a user who is trusted and whose name means that the value
3889 of the "requesting-user-name", which MUST be present, is treated as the authenticated name.
3890

3891 Note: Case "f" is intended for a tightly coupled gateway and server to work together so that the "user"
3892 name is able to be that of the gateway client and not that of the gateway. Because most, if not all, system
3893 vendors will initially implement IPP via a gateway into their existing print system, this mechanism is
3894 necessary unless the authentication mechanism allows a gateway (client) to act on behalf of some other
3895 client.

3896 The user-name has two forms:

- 3897 - one that is human readable: it is held in the MANDATORY "job-originating-user-name" Job
3898 Description attribute which is set during the job creation operations. It is used for presentation
3899 only, such as returning in queries or printing on start sheets
- 3900 - one for authorization: it is held in an undefined (by IPP) Job object attribute which is set by the job
3901 creation operation. It is used to authorize other operations, such as Send-Document, Send-URI,
3902 Cancel-Job, to determine the user when the my-jobs' attribute is specified with Get-Jobs, and to
3903 limit what attributes and values to return with Get-Job-Attributes and Get-Jobs.

3904 The human readable user name:

- 3906 - is the value of the "requesting-user-name" for cases b, d and f.
- 3907 - comes from the authentication mechanism for case e
- 3908 - is some anonymous name, such as "anonymous" for cases a and c.

3909 The user name used for authorization:

- 3911 - is the value of the "requesting-user-name" for cases b and f.
- 3912 - comes from the authentication mechanism for cases c, d and e
- 3913 - is some anonymous name, such as "anonymous" for case a.

3914 The essence of these rules for resolving conflicting sources of user-names is that a printer implementation
3915 is free to pick either source as long as it achieves consistent results. That is, if a user uses the same path
3916 for a series of requests, the requests **MUST** appear to come from the same user from the standpoint of
3917 both the human-readable user name and the user name for authorization. This rule **MUST** continue to
3918 apply even if a request could be authenticated by two or more mechanisms. It doesn't matter which of
3919 several authentication mechanisms a Printer uses as long as it achieves consistent results. If a client uses
3920 more than one authentication mechanism, it is recommended that an administrator make all credentials
3921 resolve to the same user and user-name as much as possible.
3922

3923 8.4 Restricted Queries

3924 In many IPP operations, a client supplies a list of attributes to be returned in the response. For security
3925 reasons, an IPP object may be configured not to return all attributes (or all values) that a client requests.
3926 The job attributes returned MAY depend on whether the requesting user is the same as the user that
3927 submitted the job. The IPP object MAY even return none of the requested attributes. In such cases, the
3928 status returned is the same as if the object had returned all requested attributes. The client cannot tell by
3929 such a response whether the requested attribute was present or absent on the object.

3930 8.5 IPP Security Application Profile for TLS

3931 The IPP application profile for TLS follows the standard "Mandatory Cipher Suites" requirement as
3932 documented in the TLS specification [TLS]. Client implementations MUST NOT assume any other
3933 cipher suites are supported by an IPP Printer object.

3934 If a conforming IPP object supports TLS, it MUST implement and support the "Mandatory Cipher
3935 Suites" as specified in the TLS specification and MAY support additional cipher suites.

3936 A conforming IPP client SHOULD support TLS including the "Mandatory Cipher Suites" as specified in
3937 the TLS specification. A conforming IPP client MAY support additional cipher suites.

3938 It is possible that due to certain government export restrictions some non-compliant versions of this
3939 extension could be deployed. Implementations wishing to inter-operate with such non-compliant versions
3940 MAY offer the TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA mechanism. However, since 40
3941 bit ciphers are known to be vulnerable to attack by current technology, any client which activates a 40 bit
3942 cipher MUST NOT indicate to the user that the connection is completely secure from eavesdropping.

3943 9. References

3944 [ASCII]

3945 Coded Character Set - 7-bit American Standard Code for Information Interchange (ASCII),
3946 ANSI X3.4-1986. This standard is the specification of the US-ASCII charset.

3947 [CS-POL]

3948 H. Alvestrand, "IETF Policy on Character Sets and Languages, work in progress <draft-
3949 alvestrand-charset-policy-01.txt>, August 29, 1997.

- 3950 [HTPP]
3951 J. Barnett, K. Carter, R. DeBry, "Initial Draft - Hypertext Printing Protocol - HTPP/1.0",
3952 October 1996, <ftp://ftp.pwg.org/pub/pwg/ipp/historic/http/overview.ps.gz>
- 3953 [IANA-CON]
3954 Narte, T. and Alverstran, H.T.: Guidelines for Writing an IANA Considerations Section in RFCs,
3955 Work in Progress, draft-iesg-iana-considerations-04.txt, May 21, 1998.
- 3956 [IANA-CS]
3957 IANA Registry of Coded Character Sets: <ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets>
- 3958 [IANA-CSa]
3959 N. Freed, J. Postel: IANA CharSet Registration Procedures, Work in Progress (draft-freed-
3960 charset-reg-02.txt).
- 3961 [IANA-MT]
3962 IANA Registry of Media Types: <ftp://ftp.isi.edu/in-notes/iana/assignments/media-types/>
- 3963 [IPP-PRO]
3964 Herriot, R., Butler, S., Moore, P., Tuner, R., " Internet Printing Protocol/1.0: Protocol
3965 Specifications", draft-ietf-ipp-pro-05.txt, November, 1997.
- 3966 [IPP-RAT]
3967 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",
3968 draft-ietf-ipp-rat-02.txt, November, 1997.
- 3969 [IPP-REQ]
3970 Wright, D., "Requirements for an Internet Printing Protocol", draft-ietf-ipp-req-.txt, November,
3971 1997.
- 3972 [ISO10646-1]
3973 ISO/IEC 10646-1:1993, "Information technology -- Universal Multiple-Octet Coded Character
3974 Set (UCS) - Part 1: Architecture and Basic Multilingual Plane, JTC1/SC2."
- 3975 [ISO8859-1]
3976 ISO/IEC 8859-1:1987, "Information technology -- 8-bit One-Byte Coded Character Set - Part 1:
3977 Latin Alphabet Nr 1", 1987, JTC1/SC2.
- 3978 [ISO10175]
3979 ISO/IEC 10175 Document Printing Application (DPA), June 1996.

- 3980 [LDPA]
3981 T. Hastings, S. Isaacson, M. MacKay, C. Manros, D. Taylor, P. Zehler, "LDPA - Lightweight
3982 Document Printing Application", October 1996,
3983 ftp://ftp.pwg.org/pub/pwg/ipp/historic/ldpa/ldpa8.pdf.gz
- 3984 [P1387.4]
3985 Kirk, M. (editor), POSIX System Administration - Part 4: Printing Interfaces, POSIX 1387.4 D8,
3986 1994.
- 3987 [PSIS] Herriot, R. (editor), X/Open A Printing System Interoperability Specification (PSIS), August
3988 1995.
- 3989 [PWG]
3990 Printer Working Group, <http://www.pwg.org>.
- 3991 [RFC1035]
3992 P. Mockapetris, "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION", RFC
3993 1035, November 1987.
- 3994 [RFC1179]
3995 McLaughlin, L. III, (editor), "Line Printer Daemon Protocol" RFC 1179, August 1990.
- 3996 [RFC1630]
3997 T. Berners-Lee, "Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression
3998 of Names and Addresses of Objects on the Network as used in the World-Wide Web", RFC 1630,
3999 June 1994.
- 4000 [RFC1738]
4001 Berners-Lee, T., Masinter, L., McCahill, M. , "Uniform Resource Locators (URL)", RFC 1738,
4002 December, 1994.
- 4003 [RFC1759]
4004 Smith, R., Wright, F., Hastings, T., Zilles, S., and Gyllenskog, J., "Printer MIB", RFC 1759,
4005 March 1995.
- 4006 [RFC1766]
4007 H. Alvestrand, " Tags for the Identification of Languages", RFC 1766, March 1995.
- 4008 [RFC1903]
4009 J. Case, et.al., "Textual Conventions for Version 2 of the Simple Network Management Protocol
4010 (SNMP v2)" RFC 1903, January 1996.

- 4011 [RFC1952]
4012 P. Deutsch, "GZIP file format specification version 4.3", RFC 1952, May 1996.
- 4013 [RFC2044]
4014 F. Yergeau, "UTF-8, a transformation format of Unicode and ISO 10646", RFC 2044, October
4015 1996.
- 4016 [RFC2068]
4017 R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, "Hypertext Transfer Protocol -
4018 HTTP/1.1", RFC 2068, January 1997
- 4019 [RFC2069]
4020 J. Franks, P. Hallam-Baker, J. Hostetler, P. Leach, A. Luotonen, E. Sink, L. Stewart, "An
4021 Extension to HTTP: Digest Access Authentication", RFC-2069, Jan 1997.
- 4022 [RFC2045]
4023 N. Fried, N. Borenstein, ", Multipurpose Internet Mail Extensions (MIME) Part One: Format of
4024 Internet Message Bodies " RFC 2045, November 1996.
- 4025 [RFC2046]
4026 Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. N. Freed & N.
4027 Borenstein. November 1996. (Obsoletes RFC1521, RFC1522, RFC1590), RFC 2046.
- 4028 [RFC2048]
4029 Multipurpose Internet Mail Extension (MIME) Part Four: Registration Procedures. N. Freed, J.
4030 Klensin & J. Postel. November 1996. (Format: TXT=45033 bytes) (Obsoletes RFC1521,
4031 RFC1522, RFC1590) (Also BCP0013), RFC 2048.
- 4032 [RFC2119]
4033 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 , March
4034 1997
- 4035 [SWP]
4036 P. Moore, B. Jahromi, S. Butler, "Simple Web Printing SWP/1.0", May 7, 1997,
4037 ftp://ftp.pwg.org/pub/pwg/ipp/new_PRO/swp9705.pdf

4038 10. Copyright Notice

4039 This document and translations of it may be copied and furnished to others, and derivative works that
4040 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published

4041 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright
4042 notice and this paragraph are included on all such copies and derivative works. However, this document
4043 itself may not be modified in any way, such as by removing the copyright notice or references to the
4044 Internet Society or other Internet organizations, except as needed for the purpose of developing Internet
4045 standards in which case the procedures for copyrights defined in the Internet Standards process must be
4046 followed, or as required to translate it into languages other than English.

4047 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its
4048 successors or assigns.

4049 This document and the information contained herein is provided on an "AS IS" basis and THE
4050 INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL
4051 WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
4052 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY
4053 RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
4054 PARTICULAR PURPOSE.

4055 11. Author's Address

4056 Scott A. Isaacson (Editor)
4057 Novell, Inc.
4058 122 E 1700 S
4059 Provo, UT 84606

4060
4061 Phone: 801-861-7366
4062 Fax: 801-861-2517
4063 e-mail: sisaacson@novell.com

4064
4065 Tom Hastings
4066 Xerox Corporation
4067 701 S. Aviation Blvd.
4068 El Segundo, CA 90245

4069
4070 Phone: 310-333-6413
4071 Fax: 310-333-5514
4072 e-mail: hastings@cp10.es.xerox.com

4073
4074 Robert Herriot
4075 Sun Microsystems Inc.

4076 901 San Antonio.Road, MPK-17
4077 Palo Alto, CA 94303
4078
4079 Phone: 650-786-8995
4080 Fax: 650-786-7077
4081 e-mail: robert.herriot@eng.sun.com

4082
4083 Roger deBry
4084 HUC/003G
4085 IBM Corporation
4086 P.O. Box 1900
4087 Boulder, CO 80301-9191

4088
4089 Phone: (303) 924-4080
4090 Fax: (303) 924-9889
4091 e-mail: debry@vnet.ibm.com

4092
4093 Patrick Powell
4094 San Diego State University
4095 9475 Chesapeake Dr., Suite D
4096 San Diego, CA 95123

4097
4098 Phone: (619) 874-6543
4099 Fax: (619) 279-8424
4100 e-mail: papowell@sdsu.edu

4101
4102 IPP Mailing List: ipp@pwg.org
4103 IPP Mailing List Subscription: ipp-request@pwg.org
4104 IPP Web Page: <http://www.pwg.org/ipp/>

4105
4106 Implementers of this specification are encouraged to join IPP Mailing List in order to participate in any
4107 discussions of clarification issues and review of registration proposals for additional attributes and values.

4108
4109 Other Participants:

4110 Chuck Adams - Tektronix
4111 Jeff Barnett - IBM
4112 Ron Bergman - Dataproducts Corp.
4113 Sylvan Butler, HP
4114 Keith Carter, IBM Corporation
4115 Jeff Copeland - QMS

4116 Andy Davidson - Tektronix
4117 Mabry Dozier - QMS
4118 Lee Farrell - Canon Information Systems
4119 Steve Gebert - IBM
4120 Babek Jahromi, Microsoft
4121 David Kellerman - Northlake Software
4122 Rick Landau - Digital
4123 Greg LeClair - Epson
4124 Harry Lewis - IBM
4125 Pete Loya - HP
4126 Ray Lutz - Cognisys
4127 Mike MacKay, Novell, Inc.
4128 Carl-Uno Manros, Xerox, Corp.
4129 Jay Martin - Underscore
4130 Stan McConnell - Xerox
4131 Ira McDonald, High North Inc.
4132 Paul Moore, Microsoft
4133 Tetsuya Morita - Ricoh
4134 Yuichi Niwa - Ricoh
4135 Pat Nogay - IBM
4136 Ron Norton - Printronics
4137 Bob Pentecost - HP
4138 Rob Rhoads - Intel
4139 Xavier Riley - Xerox, Corp.
4140 David Roach - Unisys
4141 Stuart Rowley, Kyocera
4142 Hiroyuki Sato - Canon
4143 Bob Setterbo - Adobe
4144 Devon Taylor, Novell, Inc.
4145 Mike Timperman - Lexmark
4146 Randy Turner - Sharp
4147 Atsushi Yuki - Kyocera
4148 Rick Yardumian - Xerox, Corp.
4149 Lloyd Young - Lexmark
4150 Bill Wagner - DPI
4151 Jim Walker - DAZEL
4152 Chris Wellens - Interworking Labs
4153 Rob Whittle - Novell
4154 Don Wright - Lexmark
4155 Peter Zehler, Xerox, Corp.

4156

Steve Zilles, Adobe

4157

4158 12. Formats for IPP Registration Proposals

4159 This section specifies the required information and the formats for proposing registrations of extensions
4160 to IPP as provided in Section 6 for:

4161

4162 1. type2 'keyword' attribute values

4163 2. type3 'keyword' attribute values

4164 3. type2 'enum' attribute values

4165 4. type3 'enum' attribute values

4166 5. attributes

4167 6. attribute syntaxes

4168 7. operations

4169 8. status codes

4170 12.1 Type2 keyword attribute values registration

4171 Type of registration: type2 keyword attribute value

4172 Name of attribute to which this keyword specification is to be added:

4173 Proposed keyword name of this keyword value:

4174 Specification of this keyword value (follow the style of IPP Model Section 4.1.3):

4175 Name of proposer:

4176 Address of proposer:

4177 Email address of proposer:

4178

4179 Note: For type2 keywords, the Designated Expert will be the point of contact for the approved
4180 registration specification, if any maintenance of the registration specification is needed.

4181 12.2 Type3 keyword attribute values registration

4182 Type of registration: type3 keyword attribute value

4183 Name of attribute to which this keyword specification is to be added:

4184 Proposed keyword name of this keyword value:

4185 Specification of this keyword value (follow the style of IPP Model Section 4.1.3):

4186 Name of proposer:

4187 Address of proposer:

4188 Email address of proposer:

4189

4190 Note: For type3 keywords, the proposer will be the point of contact for the approved registration
4191 specification, if any maintenance of the registration specification is needed.

4192 12.3 Type2 enum attribute values registration

4193 Type of registration: type2 enum attribute value
4194 Name of attribute to which this enum specification is to be added:
4195 Keyword symbolic name of this enum value:
4196 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):
4197 Specification of this enum value (follow the style of IPP Model Section 4.1.4):
4198 Name of proposer:
4199 Address of proposer:
4200 Email address of proposer:
4201

4202 Note: For type2 enums, the Designated Expert will be the point of contact for the approved registration
4203 specification, if any maintenance of the registration specification is needed.

4204 12.4 Type3 enum attribute values registration

4205 Type of registration: type3 enum attribute value
4206 Name of attribute to which this enum specification is to be added:
4207 Keyword symbolic name of this enum value:
4208 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):
4209 Specification of this enum value (follow the style of IPP Model Section 4.1.4):
4210 Name of proposer:
4211 Address of proposer:
4212 Email address of proposer:
4213

4214 Note: For type3 enums, the proposer will be the point of contact for the approved registration
4215 specification, if any maintenance of the registration specification is needed.

4216 12.5 Attribute registration

4217 Type of registration: attribute
4218 Proposed keyword name of this attribute:
4219 Types of attribute (Operation, Job Template, Job Description, Printer Description):
4220 Operations to be used with if the attribute is an operation attribute:
4221 Object (Job, Printer, etc. if bound to an object):
4222 Attribute syntax(es) (include 1setOf and range as in Section 4.2):
4223 If attribute syntax is 'keyword' or 'enum', is it type2 or type3:
4224 Specification of this attribute (follow the style of IPP Model Section 4.2):

4225 Name of proposer:

4226 Address of proposer:

4227 Email address of proposer:

4228

4229 Note: For attributes, the IPP Designated Expert will be the point of contact for the approved registration
4230 specification, if any maintenance of the registration specification is needed.

4231 12.6 Attribute Syntax registration

4232 Type of registration: attribute syntax

4233 Proposed name of this attribute syntax:

4234 Type of attribute syntax (integer, octetString, character-string, see [IPP-PRO]):

4235 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

4236 Specification of this attribute (follow the style of IPP Model Section 4.1):

4237 Name of proposer:

4238 Address of proposer:

4239 Email address of proposer:

4240

4241 Note: For attribute syntaxes, the IPP Designated Expert will be the point of contact for the approved
4242 registration specification, if any maintenance of the registration specification is needed.

4243 12.7 Operation registration

4244 Type of registration: operation

4245 Proposed name of this operation:

4246 Numeric operation-id value (to be assigned by the IPP Designated Expert in consultation with IANA):

4247 Object Target (Job, Printer, etc. that operation is upon):

4248 Specification of this attribute (follow the style of IPP Model Section 3):

4249 Name of proposer:

4250 Address of proposer:

4251 Email address of proposer:

4252

4253 Note: For operations, the IPP Designated Expert will be the point of contact for the approved
4254 registration specification, if any maintenance of the registration specification is needed.

4255 12.8 Status code registration

4256 Type of registration: status code

4257 Keyword symbolic name of this status code value:

4258 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

4259 Operations that this status code may be used with:

4260 Specification of this status code (follow the style of IPP Model Section 14 APPENDIX B: Status Codes
4261 and Suggested Status Code Messages):

4262 Name of proposer:

4263 Address of proposer:

4264 Email address of proposer:

4265

4266 Note: For status codes, the Designated Expert will be the point of contact for the approved registration
4267 specification, if any maintenance of the registration specification is needed.

4268 13. APPENDIX A: Terminology

4269 This specification uses the terminology defined in this section.

4270 13.1 Conformance Terminology

4271 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",
4272 "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be
4273 interpreted as described in RFC 2119 [RFC2119]. The sections below reiterate these definitions and
4274 include some additional ones.

4275 13.1.1 MUST

4276 This word, or the terms "REQUIRED", "SHALL" or "MANDATORY", means that the definition is an
4277 absolute requirement of the specification.

4278 13.1.2 MUST NOT

4279 This phrase, or the phrase "SHALL NOT", means that the definition is an absolute prohibition of the
4280 specification.

4281 13.1.3 SHOULD

4282 This word, or the adjective "RECOMMENDED", means that there may exist valid reasons in particular
4283 circumstances to ignore a particular item, but the full implications must be understood and carefully
4284 weighed before choosing a different course.

4285 13.1.4 SHOULD NOT

4286 This phrase, or the phrase "NOT RECOMMENDED" means that there may exist valid reasons in
4287 particular circumstances when the particular behavior is acceptable or even useful, but the full
4288 implications should be understood and the case carefully weighed before implementing any behavior
4289 described with this label.

4290 13.1.5 MAY

4291 This word, or the adjective "OPTIONAL", means that an item is truly optional. One vendor may choose
4292 to include the item because a particular marketplace requires it or because the vendor feels that it
4293 enhances the product while another vendor may omit the same item. An implementation which does not
4294 include a particular option MUST be prepared to inter-operate with another implementation which does
4295 include the option, though perhaps with reduced functionality. In the same vein an implementation which
4296 does include a particular option MUST be prepared to inter-operate with another implementation which
4297 does not include the option (except, of course, for the feature the option provides.)

4298 13.1.6 NEED NOT

4299 The verb "NEED NOT" indicates an action that the subject of the sentence does not have to implement in
4300 order to claim conformance to the standard. The verb "NEED NOT" is used instead of "MAY NOT"
4301 since "MAY NOT" sounds like a prohibition.

4302 13.2 Model Terminology

4303 13.2.1 Keyword

4304 Keywords are used within this document as identifiers of semantic entities within the abstract model (see
4305 section 4.1.3). Attribute names, some attribute values, attribute syntaxes, and attribute group names are
4306 represented as keywords.

4307 13.2.2 Attributes

4308 An attribute is an item of information that is associated with an instance of an IPP object. An attribute
4309 consists of an attribute name and one or more attribute values. Each attribute has a specific attribute
4310 syntax. All object attributes are defined in section 4 and all operation attributes are defined in section 3.

4311 Job Template Attributes are described in section 4.2. The client optionally supplies Job Template
4312 attributes in a create request (operation requests that create Job objects). The Printer object has
4313 associated attributes which define supported and default values for the Printer.

4314 13.2.2.1 Attribute Name

4315 Each attribute is uniquely identified in this document by its attribute name. An attribute name is a
4316 keyword. The keyword attribute name is given in the section header describing that attribute. In running
4317 text in this document, attribute names are indicated inside double quotation marks (") where the
4318 quotation marks are not part of the keyword itself.

4319 13.2.2.2 Attribute Group Name

4320 Related attributes are grouped into named groups. The name of the group is a keyword. The group
4321 name may be used in place of naming all the attributes in the group explicitly. Attribute groups are
4322 defined in section 3.

4323 13.2.2.3 Attribute Value

4324 Each attribute has one or more values. Attribute values are represented in the syntax type specified for
4325 that attribute. In running text in this document, attribute values are indicated inside single quotation
4326 marks ('), whether their attribute syntax is keyword, integer, text, etc. where the quotation marks are not
4327 part of the value itself.

4328 13.2.2.4 Attribute Syntax

4329 Each attribute is defined using an explicit syntax type. In this document, each syntax type is defined as a
4330 keyword with specific meaning. The protocol specification document [IPP-PRO] indicates the actual
4331 "on-the-wire" encoding rules for each syntax type. Attribute syntax types are defined in section 4.1.

4332 13.2.3 Supports

4333 By definition, a Printer object supports an attribute only if that Printer object responds with the
4334 corresponding attribute populated with some value(s) in a response to a query for that attribute. A
4335 Printer object supports an attribute value if the value is one of the Printer object's "supported values"
4336 attributes. The device behind a Printer object may exhibit a behavior that corresponds to some IPP
4337 attribute, but if the Printer object, when queried for that attribute, doesn't respond with the attribute, then
4338 as far as IPP is concerned, that implementation does not support that feature. If the Printer object's "xxx-
4339 supported" attribute is not populated with a particular value (even if that value is a legal value for that
4340 attribute), then that Printer object does not support that particular value.

4341 A conforming implementation SHALL support all MANDATORY attributes. However, even for
4342 MANDATORY attributes, conformance to IPP does not mandate that all implementations support all
4343 possible values representing all possible job processing behaviors and features. For example, if a given

4344 instance of a Printer supports only certain document formats, then that Printer responds with the
4345 "document-format-supported" attribute populated with a set of values, possibly only one, taken from the
4346 entire set of possible values defined for that attribute. This limited set of values represents the Printer's set
4347 of supported document formats. Supporting an attribute and some set of values for that attribute enables
4348 IPP end users to be aware of and make use of those features associated with that attribute and those
4349 values. If an implementation chooses to not support an attribute or some specific value, then IPP end
4350 users would have no ability to make use of that feature within the context of IPP itself. However, due to
4351 existing practice and legacy systems which are not IPP aware, there might be some other mechanism
4352 outside the scope of IPP to control or request the "unsupported" feature (such as embedded instructions
4353 within the document data itself).

4354 For example, consider the "finishings-supported" attribute.

- 4355 1) If a Printer object is not physically capable of stapling, the "finishings-supported" attribute MUST
4356 NOT be populated with the value of 'staple'.
- 4357 2) A Printer object is physically capable of stapling, however an implementation chooses not to
4358 support stapling in the IPP "finishings" attribute. In this case, 'staple' SHALL NOT be a value in
4359 the "finishings-supported" Printer object attribute. Without support for the value 'staple', an IPP
4360 end user would have no means within the protocol itself to request that a Job be stapled.
4361 However, an existing document data formatter might be able to request that the document be
4362 stapled directly with an embedded instruction within the document data. In this case, the IPP
4363 implementation does not "support" stapling, however the end user is still able to have some
4364 control over the stapling of the completed job.
- 4365 3) A Printer object is physically capable of stapling, and an implementation chooses to support
4366 stapling in the IPP "finishings" attribute. In this case, 'staple' SHALL be a value in the "finishings-
4367 supported" Printer object attribute. Doing so, would enable end users to be aware of and make
4368 use of the stapling feature using IPP attributes.

4369
4370 Even though support for Job Template attributes by a Printer object is OPTIONAL, it is
4371 RECOMMENDED that if the device behind a Printer object is capable of realizing any feature or
4372 function that corresponds to an IPP attribute and some associated value, then that implementation
4373 SHOULD support that IPP attribute and value.

4374 The set of values in any of the supported value attributes is set (populated) by some administrative
4375 process or automatic sensing mechanism that is outside the scope of IPP. For administrative policy and
4376 control reasons, an administrator may choose to make only a subset of possible values visible to the end
4377 user. In this case, the real output device behind the IPP Printer abstraction may be capable of a certain
4378 feature, however an administrator is specifying that access to that feature not be exposed to the end user
4379 through the IPP protocol. Also, since a Printer object may represent a logical print device (not just a
4380 physical device) the actual process for supporting a value is undefined and left up to the implementation.

4381 However, if a Printer object supports a value, some manual human action may be needed to realize the
4382 semantic action associated with the value, but no end user action is required.

4383 For example, if one of the values in the "finishings-supported" attribute is 'staple', the actual process
4384 might be an automatic staple action by a physical device controlled by some command sent to the device.
4385 Or, the actual process of stapling might be a manual action by an operator at an operator attended Printer
4386 object.

4387 For another example of how supported attributes function, consider a system administrator who desires
4388 to control all print jobs so that no job sheets are printed in order to conserve paper. To force no job
4389 sheets, the system administrator sets the only supported value for the "job-sheets-supported" attribute to
4390 'none'. In this case, if a client requests anything except 'none', the create request is rejected or the "job-
4391 sheets" value is ignored (depending on the value of "ipp-attribute-fidelity"). To force the use of job
4392 start/end sheets on all jobs, the administrator does not include the value 'none' in the "job-sheets-
4393 supported" attribute. In this case, if a client requests 'none', the create request is rejected or the "job-
4394 sheets" value is ignored (again depending on the value of "ipp-attribute-fidelity").

4395 13.2.4 print-stream page

4396 A "print-stream page" is a page according to the definition of pages in the language used to express the
4397 document data.

4398 13.2.5 impression

4399 An "impression" is the image (possibly many print-stream pages in different configurations) imposed onto
4400 a single media page.

4401 14. APPENDIX B: Status Codes and Suggested Status Code Messages

4402 This section defines status code enum keywords and values that are used to provide semantic information
4403 on the results of an operation request. Each operation response **MUST** include a status code. The
4404 response **MAY** also contain a status message that provides a short textual description of the status. The
4405 status code is intended for use by automata, and the status message is intended for the human end user.
4406 Since the status message is an **OPTIONAL** component of the operation response, an IPP application (i.e.,
4407 a browser, GUI, print driver or gateway) is **NOT REQUIRED** to examine or display the status message,
4408 since it **MAY** not be returned to the application.

4409 The prefix of the status keyword defines the class of response as follows:

4410 "informational" - Request received, continuing process
4411 "successful" - The action was successfully received, understood, and accepted
4412 "redirection" - Further action must be taken in order to complete the request
4413 "client-error" - The request contains bad syntax or cannot be fulfilled
4414 "server-error" - The IPP object failed to fulfill an apparently valid request
4415

4416 As with type2 enums, IPP status codes are extensible. IPP clients are NOT REQUIRED to understand
4417 the meaning of all registered status codes, though such understanding is obviously desirable. However,
4418 IPP clients SHALL understand the class of any status code, as indicated by the prefix, and treat any
4419 unrecognized response as being equivalent to the first status code of that class, with the exception that an
4420 unrecognized response SHALL NOT be cached. For example, if an unrecognized status code of "client-
4421 error-xxx-yyy" is received by the client, it can safely assume that there was something wrong with its
4422 request and treat the response as if it had received a "client-error-bad-request" status code. In such
4423 cases, IPP applications SHOULD present the OPTIONAL message (if present) to the end user since the
4424 message is likely to contain human readable information which will help to explain the unusual status.
4425 The name of the enum is the suggested status message for US English.

4426 The status code values range from 0x0000 to 0x7FFF. The value ranges for each status code class are as
4427 follows:

4428 "successful" - 0x0000 to 0x00FF
4429 "informational" - 0x0100 to 0x01FF
4430 "redirection" - 0x0200 to 0x02FF
4431 "client-error" - 0x0400 to 0x04FF
4432 "server-error" - 0x0500 to 0x05FF
4433

4434 The top half (128 values) of each range (0x0n40 to 0x0nFF, for n = 0 to 5) is reserved for private use
4435 within each status code class. Values 0x0600 to 0x7FFF are reserved for future assignment and SHALL
4436 NOT be used.

4437 14.1 Status Codes

4438 Each status code is described below. Section 14.2 contains a table that indicates which status codes apply
4439 to which operations. Sections 16.3 and 16.4 describe the suggested steps for processing IPP attributes
4440 for all operations, including returning status codes.

4441 14.1.1 Informational

4442 This class of status code indicates a provisional response and is to be used for informational purposes
4443 only.

4444 There are no status codes defined in IPP/1.0 for this class of status code.

4445 14.1.2 Successful Status Codes

4446 This class of status code indicates that the client's request was successfully received, understood, and
4447 accepted.

4448 14.1.2.1 successful-ok (0x0000)

4449 The request has succeeded. In the case of a response to a create request, the 'successful-ok' status code
4450 indicates that the request was successfully received and validated, and that the Job object has been
4451 created; it does not indicate that the job has been processed. The transition of the Job object into the
4452 'completed' state is the only indicator that the job has been printed.

4453 14.1.2.2 successful-ok-ignored-or-substituted-attributes (0x0001)

4454 The request has succeeded, but some attributes were ignored or unsupported values were substituted
4455 with supported values in order to process the job without rejecting it.

4456 14.1.2.3 successful-ok-conflicting-attributes (0x0002)

4457 The request has succeeded, but some attribute values conflicted with the values of other attributes. These
4458 conflicting values were either (1) substituted with (supported) values or (2) the attributes were removed
4459 in order to process the job without rejecting it.

4460 14.1.3 Redirection Status Codes

4461 This class of status code indicates that further action needs to be taken to fulfill the request.

4462 There are no status codes defined in IPP/1.0 for this class of status code.

4463 14.1.4 Client Error Status Codes

4464 This class of status code is intended for cases in which the client seems to have erred. The IPP object
4465 SHOULD return a message containing an explanation of the error situation and whether it is a temporary
4466 or permanent condition.

4467 14.1.4.1 client-error-bad-request (0x0400)

4468 The request could not be understood by the IPP object due to malformed syntax (such as the value of a
4469 fixed length attribute whose length does not match the prescribed length for that attribute - see section
4470 16.3). The IPP application SHOULD NOT repeat the request without modifications.

4471 14.1.4.2 client-error-forbidden (0x0401)

4472 The IPP object understood the request, but is refusing to fulfill it. Additional authentication information
4473 or authorization credentials will not help and the request SHOULD NOT be repeated. This status code is
4474 commonly used when the IPP object does not wish to reveal exactly why the request has been refused or
4475 when no other response is applicable.

4476 14.1.4.3 client-error-not-authenticated (0x0402)

4477 The request requires user authentication. The IPP client may repeat the request with suitable
4478 authentication information. If the request already included authentication information, then this status
4479 code indicates that authorization has been refused for those credentials. If this response contains the
4480 same challenge as the prior response, and the user agent has already attempted authentication at least
4481 once, then the response message may contain relevant diagnostic information. This status codes reveals
4482 more information than "client-error-forbidden".

4483 14.1.4.4 client-error-not-authorized (0x0403)

4484 The requester is not authorized to perform the request. Additional authentication information or
4485 authorization credentials will not help and the request SHOULD NOT be repeated. This status code is
4486 used when the IPP object wishes to reveal that the authentication information is understandable, however,
4487 the requester is explicitly not authorized to perform the request. This status codes reveals more
4488 information than "client-error-forbidden" and "client-error-not-authenticated".

4489 14.1.4.5 client-error-not-possible (0x0404)

4490 This status code is used when the request is for something that can not happen. For example, there might
4491 be a request to cancel a job that has already been canceled or aborted by the system. The IPP client
4492 SHOULD NOT repeat the request.

4493 14.1.4.6 client-error-timeout (0x0405)

4494 The client did not produce a request within the time that the IPP object was prepared to wait. For
4495 example, a client issued a Create-Job operation and then, after a long period of time, issued a Send-

4496 Document operation and this error status code was returned in response to the Send-Document request
4497 (see section 3.3.1). The IPP object might have been forced to clean up resources that had been held for
4498 the waiting additional Documents. The IPP object was forced to close the Job since the client took too
4499 long. The client SHOULD NOT repeat the request without modifications.

4500 14.1.4.7 client-error-not-found (0x0406)

4501 The IPP object has not found anything matching the request URI. No indication is given of whether the
4502 condition is temporary or permanent. For example, a client with an old reference to a Job (a URI) tries to
4503 cancel the Job, however in the mean time the Job might have been completed and all record of it at the
4504 Printer has been deleted. This status code, 'client-error-not-found' is returned indicating that the
4505 referenced Job can not be found. This error status code is also used when a client supplies a URI as a
4506 reference to the document data in either a Print-URI or Send-URI operation, but the document can not
4507 be found.

4508 In practice, an IPP application should avoid a not found situation by first querying and presenting a list of
4509 valid Printer URIs and Job URIs to the end-user.

4510 14.1.4.8 client-error-gone (0x0407)

4511 The requested object is no longer available and no forwarding address is known. This condition should
4512 be considered permanent. Clients with link editing capabilities should delete references to the request
4513 URI after user approval. If the IPP object does not know or has no facility to determine, whether or not
4514 the condition is permanent, the status code "client-error-not-found" should be used instead.

4515 This response is primarily intended to assist the task of maintenance by notifying the recipient that the
4516 resource is intentionally unavailable and that the IPP object administrator desires that remote links to that
4517 resource be removed. It is not necessary to mark all permanently unavailable resources as "gone" or to
4518 keep the mark for any length of time -- that is left to the discretion of the IPP object administrator.

4519 14.1.4.9 client-error-request-entity-too-large (0x0408)

4520 The IPP object is refusing to process a request because the request entity is larger than the IPP object is
4521 willing or able to process. An IPP Printer returns this status code when it limits the size of print jobs and
4522 it receives a print job that exceeds that limit or when the attributes are so many that their encoding causes
4523 the request entity to exceed IPP object capacity.

4524 14.1.4.10 client-error-request-value-too-long (0x0409)

4525 The IPP object is refusing to service the request because one or more of the client client-supplied
4526 attributes has a variable length value that is longer than the maximum length specified for that attribute.
4527 The IPP object might not have sufficient resources (memory, buffers, etc.) to process (even temporarily),
4528 interpret, and/or ignore the large value. Another use of this error code is when the IPP object supports
4529 the processing of the large value, but during the processing of the request as a whole, the object may pass
4530 the value onto some other system component which is not able to accept the large value. For more
4531 details, see section 16.3.

4532 Note: For attribute values that are URIs, this rare condition is only likely to occur when a client has
4533 improperly submitted a request with long query information (e.g. an IPP application allows an end-user to
4534 enter an invalid URI), when the client has descended into a URI "black hole" of redirection (e.g., a
4535 redirected URI prefix that points to a suffix of itself), or when the IPP object is under attack by a client
4536 attempting to exploit security holes present in some IPP objects using fixed-length buffers for reading or
4537 manipulating the Request-URI.

4538 14.1.4.11 client-error-document-format-not-supported (0x040A)

4539 The IPP object is refusing to service the request because the document data is in a format, as specified in
4540 the "document-format" operation attribute, that is not supported by the Printer object. This error is
4541 returned independent of the client-supplied "ipp-attribute-fidelity". The Printer object SHALL return this
4542 status code, even if there are other attributes that are not supported as well, since this error is a bigger
4543 problem than with Job Template attributes.

4544 14.1.4.12 client-error-attributes-or-values-not-supported (0x040B)

4545 In a create request, if the Printer object does not support one or more attributes or attribute values
4546 supplied in the request and the client supplied the "ipp-attributes-fidelity" operation attribute with the
4547 'true' value, the Printer object shall return this status code. For example, if the request indicates 'iso-a4'
4548 media, but that media type is not supported by the Printer object. Or, if the client supplies an optional
4549 attribute and the attribute itself is not even supported by the Printer. If the "ipp-attribute-fidelity"
4550 attribute is 'false', the Printer SHALL ignore or substitute values for unsupported attributes and values
4551 rather than reject the request and return this status code.

4552 For any operation where a client requests attributes (such as a Get-Jobs, Get-Printer-Attributes, or Get-
4553 Job-Attributes operation), if the IPP object does not support one or more of the requested attributes, the
4554 IPP object simply ignores the unsupported requested attributes and processes the request as if they had
4555 not been supplied, rather than returning this status code.

4556 14.1.4.13 client-error-uri-scheme-not-supported (0x040C)

4557 The type of the client supplied URI in a Print-URI or a Send-URI operation is not supported.

4558 14.1.4.14 client-error-charset-not-supported (0x040D)

4559 For any operation, if the IPP Printer does not support the charset supplied by the client in the "attributes-
4560 charset" operation attribute, the Printer SHALL reject the operation and return this status (see Section
4561 3.1.4.1).

4562 14.1.4.15 client-error-conflicting-attributes (0x040E)

4563 The request is rejected because some attribute values conflicted with the values of other attributes.

4564 14.1.5 Server Error Status Codes

4565 This class of status codes indicates cases in which the IPP object is aware that it has erred or is incapable
4566 of performing the request. The IPP object SHOULD include a message containing an explanation of the
4567 error situation, and whether it is a temporary or permanent condition.

4568 14.1.5.1 server-error-internal-error (0x0500)

4569 The IPP object encountered an unexpected condition that prevented it from fulfilling the request. This
4570 error status code differs from "server-error-temporary-error" in that it implies a more permanent type of
4571 internal error. It also differs from "server-error-device-error" in that it implies an unexpected condition
4572 (unlike a paper-jam or out-of-toner problem which is undesirable but expected). This error status code
4573 indicates that probably some knowledgeable human intervention is required.

4574 14.1.5.2 server-error-operation-not-supported (0x0501)

4575 The IPP object does not support the functionality required to fulfill the request. This is the appropriate
4576 response when the IPP object does not recognize an operation or is not capable of supporting it.

4577 14.1.5.3 server-error-service-unavailable (0x0502)

4578 The IPP object is currently unable to handle the request due to a temporary overloading or maintenance
4579 of the IPP object. The implication is that this is a temporary condition which will be alleviated after some
4580 delay. If known, the length of the delay may be indicated in the message. If no delay is given, the IPP
4581 application should handle the response as it would for a "server-error-temporary-error" response. If the

4582 condition is more permanent, the error status codes "client-error-gone" or "client-error-not-found" could
4583 be used.

4584 14.1.5.4 server-error-version-not-supported (0x0503)

4585 The IPP object does not support, or refuses to support, the IPP protocol version that was used in the
4586 request message. The IPP object is indicating that it is unable or unwilling to complete the request using
4587 the same version as supplied in the request other than with this error message. The response should
4588 contain a Message describing why that version is not supported and what other versions are supported by
4589 that IPP object.

4590 A conforming IPP/1.0 client SHALL specify the valid version ('1.0') on each request. A conforming
4591 IPP/1.0 object SHALL NOT return this status code to a conforming IPP/1.0 client. An IPP object
4592 SHALL return this status code to a non-conforming IPP client. The response SHALL identify in the
4593 "version-number" operation attribute the closest version number that the IPP object does support.

4594 14.1.5.5 server-error-device-error (0x0504)

4595 A printer error, such as a paper jam, occurs while the IPP object processes a Print or Send operation.
4596 The response contains the true Job Status (the values of the "job-state" and "job-state-reasons"
4597 attributes). Additional information can be returned in the optional "job-state-message" attribute value or
4598 in the OPTIONAL status message that describes the error in more detail. This error status code is only
4599 returned in situations where the Printer is unable to accept the create request because of such a device
4600 error. For example, if the Printer is unable to spool, and can only accept one job at a time, the reason it
4601 might reject a create request is that the printer currently has a paper jam. In many cases however, where
4602 the Printer object can accept the request even though the Printer has some error condition, the
4603 'successful-ok' status code will be returned. In such a case, the client would look at the returned Job
4604 Object Attributes or later query the Printer to determine its state and state reasons.

4605 14.1.5.6 server-error-temporary-error (0x0505)

4606 A temporary error such as a buffer full write error, a memory overflow (i.e. the document data exceeds
4607 the memory of the Printer), or a disk full condition, occurs while the IPP Printer processes an operation.
4608 The client MAY try the unmodified request again at some later point in time with an expectation that the
4609 temporary internal error condition may have been cleared. Alternatively, as an implementation option, a
4610 Printer object MAY delay the response until the temporary condition is cleared so that no error is
4611 returned.

4612 14.1.5.7 server-error-not-accepting-jobs (0x0506)

4613 A temporary error indicating that the Printer is not currently accepting jobs, because the administrator has
4614 set the value of the Printer's "printer-is-not-accepting-jobs" attribute to 'false' (by means outside of
4615 IPP/1.0).

4616 14.1.5.8 server-error-busy (0x0507)

4617 A temporary error indicating that the Printer is too busy processing jobs and/or other requests. The client
4618 SHOULD try the unmodified request again at some later point in time with an expectation that the
4619 temporary busy condition will have been cleared.

4620 14.2 Status Codes for IPP Operations

4621 PJ = Print-Job, PU = Print-URI, CJ = Create-Job, SD = Send-Document
 4622 SU = Send-URI, V = Validate-Job, GA = Get-Job-Attributes and
 4623 Get-Printer-Attributes, GJ = Get-Jobs, C = Cancel-Job

4624		IPP Operations								
4625	IPP Status Keyword	PJ	PU	CJ	SD	SU	V	GA	GJ	C
4626	-----	--	--	--	--	--	--	--	--	--
4627	successful-ok	x	x	x	x	x	x	x	x	x
4628	successful-ok-ignored-or-substituted-	x	x	x	x	x	x	x	x	x
4629	attributes									
4630	successful-ok-conflicting-attributes	x	x	x	x	x	x	x	x	x
4631	client-error-bad-request	x	x	x	x	x	x	x	x	x
4632	client-error-forbidden	x	x	x	x	x	x	x	x	x
4633	client-error-not-authenticated	x	x	x	x	x	x	x	x	x
4634	client-error-not-authorized	x	x	x	x	x	x	x	x	x
4635	client-error-not-possible	x	x	x	x	x	x	x	x	x
4636	client-error-timeout	x	x	x	x	x	x	x	x	x
4637	client-error-not-found	x	x	x	x	x	x	x	x	x
4638	client-error-gone	x	x	x	x	x	x	x	x	x
4639	client-error-request-entity-too-large	x	x	x	x	x	x	x	x	x
4640	client-error-request-value-too-long	x	x	x	x	x	x	x	x	x
4641	client-error-document-format-not-	x	x		x	x	x	x		
4642	supported									
4643	client-error-attributes-or-values-not-	x	x	x	x	x	x	x	x	x
4644	supported									
4645	client-error-uri-scheme-not-supported		x			x				
4646	client-error-charset-not-supported	x	x	x	x	x	x	x	x	x
4647	client-error-conflicting-attributes	x	x	x	x	x	x	x	x	x
4648	server-error-internal-error	x	x	x	x	x	x	x	x	x
4649	server-error-operation-not-supported		x	x	x	x				
4650	server-error-service-unavailable	x	x	x	x	x	x	x	x	x
4651	server-error-version-not-supported	x	x	x	x	x	x	x	x	x
4652	server-error-device-error	x	x	x	x	x				
4653	server-error-temporary-error	x	x	x	x	x				
4654	server-error-not-accepting-jobs	x	x	x	x	x	x			
4655	server-error-busy	x	x	x	x	x	x	x	x	x
4656										
4657										
4658										

4659 15. APPENDIX C: "media" keyword values

4660 Standard keyword values are taken from several sources.

4661 Standard values are defined (taken from DPA[ISO10175] and the Printer MIB[RFC1759]):

4662 'default': The default medium for the output device
4663 'iso-a4-white': Specifies the ISO A4 white medium
4664 'iso-a4-colored': Specifies the ISO A4 colored medium
4665 'iso-a4-transparent' Specifies the ISO A4 transparent medium
4666 'iso-a3-white': Specifies the ISO A3 white medium
4667 'iso-a3-colored': Specifies the ISO A3 colored medium
4668 'iso-a5-white': Specifies the ISO A5 white medium
4669 'iso-a5-colored': Specifies the ISO A5 colored medium
4670 'iso-b4-white': Specifies the ISO B4 white medium
4671 'iso-b4-colored': Specifies the ISO B4 colored medium
4672 'iso-b5-white': Specifies the ISO B5 white medium
4673 'iso-b5-colored': Specifies the ISO B5 colored medium
4674 'jis-b4-white': Specifies the JIS B4 white medium
4675 'jis-b4-colored': Specifies the JIS B4 colored medium
4676 'jis-b5-white': Specifies the JIS B5 white medium
4677 'jis-b5-colored': Specifies the JIS B5 colored medium
4678

4679 The following standard values are defined for North American media:

4680 'na-letter-white': Specifies the North American letter white medium
4681 'na-letter-colored': Specifies the North American letter colored medium
4682 'na-letter-transparent': Specifies the North American letter transparent medium
4683 'na-legal-white': Specifies the North American legal white medium
4684 'na-legal-colored': Specifies the North American legal colored medium
4685

4686 The following standard values are defined for envelopes:

4687 'iso-b4-envelope': Specifies the ISO B4 envelope medium
4688 'iso-b5-envelope': Specifies the ISO B5 envelope medium
4689 'iso-c3-envelope': Specifies the ISO C3 envelope medium
4690 'iso-c4-envelope': Specifies the ISO C4 envelope medium
4691 'iso-c5-envelope': Specifies the ISO C5 envelope medium
4692 'iso-c6-envelope': Specifies the ISO C6 envelope medium
4693 'iso-designated-long-envelope': Specifies the ISO Designated Long envelope medium
4694 'na-10x13-envelope': Specifies the North American 10x13 envelope medium
4695 'na-9x12-envelope': Specifies the North American 9x12 envelope medium
4696 'monarch-envelope': Specifies the Monarch envelope
4697 'na-number-10-envelope': Specifies the North American number 10 business envelope medium

4698 'na-7x9-envelope': Specifies the North American 7x9 inch envelope
4699 'na-9x11-envelope': Specifies the North American 9x11 inch envelope
4700 'na-10x14-envelope': Specifies the North American 10x14 inch envelope
4701 'na-number-9-envelope': Specifies the North American number 9 business envelope
4702 'na-6x9-envelope': Specifies the North American 6x9 inch envelope
4703 'na-10x15-envelope': Specifies the North American 10x15 inch envelope
4704

4705 The following standard values are defined for the less commonly used media (white-only):

4706 'executive-white': Specifies the white executive medium
4707 'folio-white': Specifies the folio white medium
4708 'invoice-white': Specifies the white invoice medium
4709 'ledger-white': Specifies the white ledger medium
4710 'quarto-white': Specifies the white quarto medium
4711 'iso-a0-white': Specifies the ISO A0 white medium
4712 'iso-a1-white': Specifies the ISO A1 white medium
4713 'iso-a2-white': Specifies the ISO A2 white medium
4714 'iso-a6-white': Specifies the ISO A6 white medium
4715 'iso-a7-white': Specifies the ISO A7 white medium
4716 'iso-a8-white': Specifies the ISO A8 white medium
4717 'iso-a9-white': Specifies the ISO A9 white medium
4718 'iso-10-white': Specifies the ISO A10 white medium
4719 'iso-b0-white': Specifies the ISO B0 white medium
4720 'iso-b1-white': Specifies the ISO B1 white medium
4721 'iso-b2-white': Specifies the ISO B2 white medium
4722 'iso-b3-white': Specifies the ISO B3 white medium
4723 'iso-b6-white': Specifies the ISO B6 white medium
4724 'iso-b7-white': Specifies the ISO B7 white medium
4725 'iso-b8-white': Specifies the ISO B8 white medium
4726 'iso-b9-white': Specifies the ISO B9 white medium
4727 'iso-b10-white': Specifies the ISO B10 white medium
4728 'jis-b0-white': Specifies the JIS B0 white medium
4729 'jis-b1-white': Specifies the JIS B1 white medium
4730 'jis-b2-white': Specifies the JIS B2 white medium
4731 'jis-b3-white': Specifies the JIS B3 white medium
4732 'jis-b6-white': Specifies the JIS B6 white medium
4733 'jis-b7-white': Specifies the JIS B7 white medium
4734 'jis-b8-white': Specifies the JIS B8 white medium
4735 'jis-b9-white': Specifies the JIS B9 white medium
4736 'jis-b10-white': Specifies the JIS B10 white medium

4737

4738 The following standard values are defined for engineering media:

4739 'a': Specifies the engineering A size medium

4740 'b': Specifies the engineering B size medium

4741 'c': Specifies the engineering C size medium

4742 'd': Specifies the engineering D size medium

4743 'e': Specifies the engineering E size medium

4744

4745 The following standard values are defined for input-trays (from ISO DPA and the Printer MIB):

4746 'top': The top input tray in the printer.

4747 'middle': The middle input tray in the printer.

4748 'bottom': The bottom input tray in the printer.

4749 'envelope': The envelope input tray in the printer.

4750 'manual': The manual feed input tray in the printer.

4751 'large-capacity': The large capacity input tray in the printer.

4752 'main': The main input tray

4753 'side': The side input tray

4754

4755 The following standard values are defined for media sizes (from ISO DPA):

4756 'iso-a0': Specifies the ISO A0 size: 841 mm by 1189 mm as defined in ISO 216

4757 'iso-a1': Specifies the ISO A1 size: 594 mm by 841 mm as defined in ISO 216

4758 'iso-a2': Specifies the ISO A2 size: 420 mm by 594 mm as defined in ISO 216

4759 'iso-a3': Specifies the ISO A3 size: 297 mm by 420 mm as defined in ISO 216

4760 'iso-a4': Specifies the ISO A4 size: 210 mm by 297 mm as defined in ISO 216

4761 'iso-a5': Specifies the ISO A5 size: 148 mm by 210 mm as defined in ISO 216

4762 'iso-a6': Specifies the ISO A6 size: 105 mm by 148 mm as defined in ISO 216

4763 'iso-a7': Specifies the ISO A7 size: 74 mm by 105 mm as defined in ISO 216

4764 'iso-a8': Specifies the ISO A8 size: 52 mm by 74 mm as defined in ISO 216

4765 'iso-a9': Specifies the ISO A9 size: 37 mm by 52 mm as defined in ISO 216

4766 'iso-a10': Specifies the ISO A10 size: 26 mm by 37 mm as defined in ISO 216

4767 'iso-b0': Specifies the ISO B0 size: 1000 mm by 1414 mm as defined in ISO 216

4768 'iso-b1': Specifies the ISO B1 size: 707 mm by 1000 mm as defined in ISO 216

4769 'iso-b2': Specifies the ISO B2 size: 500 mm by 707 mm as defined in ISO 216

4770 'iso-b3': Specifies the ISO B3 size: 353 mm by 500 mm as defined in ISO 216

4771 'iso-b4': Specifies the ISO B4 size: 250 mm by 353 mm as defined in ISO 216

4772 'iso-b5': Specifies the ISO B5 size: 176 mm by 250 mm as defined in ISO 216

4773 'iso-b6': Specifies the ISO B6 size: 125 mm by 176 mm as defined in ISO 216
4774 'iso-b7': Specifies the ISO B7 size: 88 mm by 125 mm as defined in ISO 216
4775 'iso-b8': Specifies the ISO B8 size: 62 mm by 88 mm as defined in ISO 216
4776 'iso-b9': Specifies the ISO B9 size: 44 mm by 62 mm as defined in ISO 216
4777 'iso-b10': Specifies the ISO B10 size: 31 mm by 44 mm as defined in ISO 216
4778 'na-letter': Specifies the North American letter size: 8.5 inches by 11 inches
4779 'na-legal': Specifies the North American legal size: 8.5 inches by 14 inches
4780 'executive': Specifies the executive size (7.25 X 10.5 in)
4781 'folio': Specifies the folio size (8.5 X 13 in)
4782 'invoice': Specifies the invoice size (5.5 X 8.5 in)
4783 'ledger': Specifies the ledger size (11 X 17 in)
4784 'quarto': Specifies the quarto size (8.5 X 10.83 in)
4785 'iso-c3': Specifies the ISO C3 size: 324 mm by 458 mm as defined in ISO 269
4786 'iso-c4': Specifies the ISO C4 size: 229 mm by 324 mm as defined in ISO 269
4787 'iso-c5': Specifies the ISO C5 size: 162 mm by 229 mm as defined in ISO 269
4788 'iso-c6': Specifies the ISO C6 size: 114 mm by 162 mm as defined in ISO 269
4789 'iso-designated-long': Specifies the ISO Designated Long size: 110 mm by 220 mm as defined in ISO
4790 269
4791 'na-10x13-envelope': Specifies the North American 10x13 size: 10 inches by 13 inches
4792 'na-9x12-envelope': Specifies the North American 9x12 size: 9 inches by 12 inches
4793 'na-number-10-envelope': Specifies the North American number 10 business envelope size: 4.125
4794 inches by 9.5 inches
4795 'na-7x9-envelope': Specifies the North American 7x9 inch envelope size
4796 'na-9x11-envelope': Specifies the North American 9x11 inch envelope size
4797 'na-10x14-envelope': Specifies the North American 10x14 inch envelope size
4798 'na-number-9-envelope': Specifies the North American number 9 business envelope size
4799 'na-6x9-envelope': Specifies the North American 6x9 envelope size
4800 'na-10x15-envelope': Specifies the North American 10x15 envelope size
4801 'monarch-envelope': Specifies the Monarch envelope size (3.87 x 7.5 in)
4802 'jis-b0': Specifies the JIS B0 size: 1030mm x 1456mm
4803 'jis-b1': Specifies the JIS B1 size: 728mm x 1030mm
4804 'jis-b2': Specifies the JIS B2 size: 515mm x 728mm
4805 'jis-b3': Specifies the JIS B3 size: 364mm x 515mm
4806 'jis-b4': Specifies the JIS B4 size: 257mm x 364mm
4807 'jis-b5': Specifies the JIS B5 size: 182mm x 257mm
4808 'jis-b6': Specifies the JIS B6 size: 128mm x 182mm
4809 'jis-b7': Specifies the JIS B7 size: 91mm x 128mm
4810 'jis-b8': Specifies the JIS B8 size: 64mm x 91mm
4811 'jis-b9': Specifies the JIS B9 size: 45mm x 64mm
4812 'jis-b10': Specifies the JIS B10 size: 32mm x 45mm

4813 16. APPENDIX D: Processing IPP Attributes

4814 When submitting a print job to a Printer object, the IPP model allows a client to supply operation and Job
4815 Template attributes along with the document data. These Job Template attributes in the create request
4816 affect the rendering, production and finishing of the documents in the job. Similar types of instructions
4817 may also be contained in the document to be printed, that is, embedded within the print data itself. In
4818 addition, the Printer has a set of attributes that describe what rendering and finishing options which are
4819 supported by that Printer. This model, which allows for flexibility and power, also introduces the
4820 potential that at job submission time, these client-supplied attributes may conflict with either:

- 4821 - what the implementation is capable of realizing (i.e., what the Printer supports), as well as
- 4822 - the instructions embedded within the print data itself.

4823
4824 The following sections describe how these two types of conflicts are handled in the IPP model.

4825 16.1 Fidelity

4826 If there is a conflict between what the client requests and what a Printer object supports, the client may
4827 request one of two possible conflict handling mechanisms:

- 4828 1) either reject the job since the job can not be processed exactly as specified, or
- 4829 2) allow the Printer to make any changes necessary to proceed with processing the Job the best it can.

4830
4831 In the first case the client is indicating to the Printer object: "Print the job exactly as specified with no
4832 exceptions, and if that can't be done, don't even bother printing the job at all." In the second case, the
4833 client is indicating to the Printer object: "It is more important to make sure the job is printed rather than
4834 be processed exactly as specified; just make sure the job is printed even if client supplied attributes need
4835 to be changed or ignored."

4836 The IPP model accounts for this situation by introducing an "ipp-attribute-fidelity" attribute.

4837 In a create request, "ipp-attribute-fidelity" is a boolean operation attribute that is OPTIONALLY
4838 supplied by the client. The value 'true' indicates that total fidelity to client supplied Job Template
4839 attributes and values is required. The client is requesting that the Job be printed exactly as specified, and
4840 if that is not possible then the job MUST be rejected rather than processed incorrectly. The value 'false'
4841 indicates that a reasonable attempt to print the Job is acceptable. If a Printer does not support some of
4842 the client supplied Job Template attributes or values, the Printer SHALL ignore them or substitute any
4843 supported value for unsupported values, respectively. The Printer may choose to substitute the default
4844 value associated with that attribute, or use some other supported value that is similar to the unsupported
4845 requested value. For example, if a client supplies a "media" value of 'na-letter', the Printer may choose to

4846 substitute 'iso-a4' rather than a default value of 'envelope'. If the client does not supply the "ipp-attribute-
4847 fidelity" attribute, the Printer assumes a value of 'false'.

4848 Each Printer implementation MUST support both types of "fidelity" printing (that is whether the client
4849 supplies a value of 'true' or 'false'):

- 4850 - If the client supplies 'false' or does not supply the attribute, the Printer object SHALL always accept
4851 the request by ignoring unsupported Job Template attributes and by substituting unsupported
4852 values of supported Job Template attributes with supported values.
- 4853 - If the client supplies 'true', the Printer object SHALL reject the request if the client supplies
4854 unsupported Job Template attributes.

4855

4856 Since a client can always query a Printer to find out exactly what is and is not supported, "ipp-attribute-
4857 fidelity" set to 'false' is useful when:

- 4858 1) The End-User uses a command line interface to request attributes that might not be supported.
- 4859 2) In a GUI context, if the End User expects the job might be moved to another printer and prefers a
4860 sub-optimal result to nothing at all.
- 4861 3) The End User just wants something reasonable in lieu of nothing at all.

4862

4863 16.2 Page Description Language (PDL) Override

4864 If there is a conflict between the value of an IPP Job Template attribute and a corresponding instruction
4865 in the document data, the value of the IPP attribute SHOULD take precedence over the document
4866 instruction. Consider the case where a previously formatted file of document data is sent to an IPP
4867 Printer. In this case, if the client supplies any attributes at job submission time, the client desires that
4868 those attributes override the embedded instructions. Consider the case were a previously formatted
4869 document has embedded in it commands to load 'iso-a4' media. However, the document is passed to an
4870 end user that only has access to a printer with 'na-letter' media loaded. That end user most likely wants to
4871 submit that document to an IPP Printer with the "media" Job Template attribute set to 'na-letter'. The job
4872 submission attribute should take precedence over the embedded PDL instruction. However, until
4873 companies that supply document data interpreters allow a way for external IPP attributes to take
4874 precedence over embedded job production instructions, a Printer might not be able to support the
4875 semantics that IPP attributes override the embedded instructions.

4876 The IPP model accounts for this situation by introducing a "pdl-override-supported" attribute that
4877 describes the Printer objects capabilities to override instructions embedded in the PDL data stream. The
4878 value of the "pdl-override-supported" attribute is configured by means outside IPP/1.0.

4879 This MANDATORY Printer attribute takes on the following values:

- 4880 - 'attempted': This value indicates that the Printer object attempts to make the IPP attribute values
4881 take precedence over embedded instructions in the document data, however there is no guarantee.
4882 - 'not-attempted': This value indicates that the Printer object makes no attempt to make the IPP
4883 attribute values take precedence over embedded instructions in the document data.
4884

4885 At job processing time, an implementation that supports the value of 'attempted' might do one of several
4886 different actions:

- 4887 1) Generate an output device specific command sequence to realize the feature represented by the IPP
4888 attribute value.
- 4889 2) Parse the document data itself and replace the conflicting embedded instruction with a new
4890 embedded instruction that matches the intent of the IPP attribute value.
- 4891 3) Indicate to the Printer that external supplied attributes take precedence over embedded instructions
4892 and then pass the external IPP attribute values to the document data interpreter.
- 4893 4) Anything else that allows for the semantics that IPP attributes override embedded document data
4894 instructions.

4895
4896 Since 'attempted' does not offer any type of guarantee, even though a given Printer object might not do a
4897 very "good" job of attempting to ensure that IPP attributes take a higher precedence over instructions
4898 embedded in the document data, it would still be a conforming implementation.

4899 At job processing time, an implementation that supports the value of 'not-attempted' might do one of the
4900 following actions:

- 4901 1) Simply pre-pend the document data with the PDL instruction that corresponds to the client-
4902 supplied PDL attribute, such that if the document data also has the same PDL instruction, it will
4903 override what the Printer object pre-pended. In other words, this implementation is using the
4904 same implementation semantics for the client-supplied IPP attributes as for the Printer object
4905 defaults.
- 4906 2) Parse the document data and replace the conflicting embedded instruction with a new embedded
4907 instruction that approximates, but does not match, the semantic intent of the IPP attribute value.
4908

4909 Note: The "ipp-attribute-fidelity" attribute applies to the Printer's ability to either accept or reject other
4910 unsupported Job Template attributes. In other words, if "ipp-attribute-fidelity" is set to 'true', a Job is
4911 accepted if and only if the client supplied Job Template attributes and values are supported by the Printer.
4912 Whether these attributes actually affect the processing of the Job when the document data contains
4913 embedded instructions depends on the ability of the Printer to override the instructions embedded in the
4914 document data with the semantics of the IPP attributes. If the document data attributes can be
4915 overridden ("pdl-override-supported" set to 'attempted'), the Printer makes an attempt to use the IPP
4916 attributes when processing the Job. If the document data attributes can not be overridden ("pdl-override-
4917 supported" set to 'not-attempted'), the Printer makes no attempt to override the embedded document data

4918 instructions with the IPP attributes when processing the Job, and hence, the IPP attributes may fail to
4919 affect the Job processing and output when the corresponding instruction is embedded in the document
4920 data.

4921 16.3 Suggested Operation Processing Steps for All Operations

4922 When an IPP object receives a request, the IPP object either accepts or rejects the request. In order to
4923 determine whether or not to accept or reject the request, the IPP object SHOULD execute the following
4924 steps. The order of the steps may be rearranged and/or combined, including making one or multiple
4925 passes over the request. Therefore, the error status codes returned may differ between implementations.
4926 The next section contains the additional steps for the Print-Job, Validate-Job, Print-URI, Create-Job,
4927 Send-Document, and Send-URI operations that create jobs, adds documents, and validates jobs.

4928 In the following, processing continues step by step until a "RETURNS the xxx status code ..." statement
4929 is encountered. Error returns are indicated by the verb: "REJECTS". Since clients have difficulty getting
4930 the status code before sending all of the document data in a Print-Job request, clients SHOULD use the
4931 Validate-Job operation before sending large documents to be printed, in order to validate whether the IPP
4932 Printer will accept the job or not.

4933 It is assumed that security authentication and authorization has already taken place at a lower layer.

4934 16.3.1 Validate version number

4935 Every request and every response contains the "version-number" attribute. The value of this attribute is
4936 the major and minor version number of the syntax and semantics that the client and IPP object is using,
4937 respectively. The "version-number" attribute remains in a fixed position across all future versions so that
4938 all clients and IPP object that support future versions can determine which version is being used. The IPP
4939 object checks to see if the major version number supplied in the request is supported. If not, the Printer
4940 object REJECTS the request and RETURNS the 'server-error-version-not-supported' status code in the
4941 response. The IPP object returns in the "version-number" response attribute the major and minor version
4942 for the error response. Thus the client can learn at least one major and minor version that the IPP object
4943 supports. The IPP object is encouraged to return the closest version number to the one supplied by the
4944 client.

4945 The checking of the minor version number is implementation dependent, however if the client supplied
4946 minor version is explicitly supported, the IPP object SHALL respond using that identical minor version
4947 number. If the requested minor version is not supported (the requested minor version is either higher or
4948 lower) than a supported minor version, the IPP object SHOULD return the closest supported minor
4949 version.

4950 16.3.2 Validate operation identifier

4951 The Printer object checks to see if the "operation-id" attribute supplied by the client is supported as
4952 indicated in the Printer object's "printer-operations-supported" attribute. If not, the Printer REJECTS the
4953 request and returns the 'server-error-operation-not-supported' status code in the response.

4954 16.3.3 Validate the request identifier

4955 The Printer object checks to see if the "request-id" attribute supplied by the client is in range. If the value
4956 is not between 1 and $2^{*}31 - 1$ (inclusive), the Printer object REJECTS the request and returns the
4957 'client-error-bad-request' status code in the response.

4958 Note: The "version-number", attribute, "operation-id", and the "request-id" attributes in the same fixed
4959 octet positions in all versions of the protocol. These fields are validated before proceeding with the rest
4960 of the validation.

4961 16.3.4 Validate attribute group and attribute presence and order

4962 The order of the following validation steps depends on implementation.

4963 16.3.4.1 Validate the presence and order of attribute groups

4964 Client requests and IPP object responses contain attribute groups that Section 3 requires to be present
4965 and in a specified order. An IPP object verifies that the attribute groups are present and in the correct
4966 order in requests supplied by clients (attribute groups without an * in the following tables).

4967 If an IPP object receives a request with (1) required attribute groups missing, or (2) the attributes groups
4968 are out of order, or (3) the groups are repeated, the IPP object REJECTS the request and RETURNS the
4969 'client-error-bad-request' status code. For example, it is an error for the Job Template Attributes group
4970 to occur before the Operation Attributes group, for the Operation Attributes group to be omitted, or for
4971 an attribute group to occur more than once, except in the Get-Jobs response.

4972 Since this kind of attribute group error is most likely to be an error detected by a client developer rather
4973 than by a customer, the IPP object NEED NOT return an indication of which attribute group was in error
4974 in either the Unsupported Attributes group or the Status Message. Also, the IPP object NEED NOT find
4975 all attribute group errors before returning this error.

4976 16.3.4.2 Ignore unknown attribute groups in the expected position

4977 Future attribute groups may be added to the specification at the end of requests just before the Document
4978 Content and at the end of response, except for the Get-Jobs response, where it maybe there or before the

4979 first job attributes returned. If an IPP object receives an unknown attribute group in these positions, it
4980 ignores the entire group, rather than returning an error, since that group may be a new group in a later
4981 minor version of the protocol that can be ignored. (If the new attribute group cannot be ignored without
4982 confusing the client, the major version number would have been increased in the protocol document and
4983 in the request). If the unknown group occurs in a different position, the IPP object REJECTS the request
4984 and RETURNS the 'client-error-bad-request' status code.

4985 Clients also ignore unknown attribute groups returned in a response.

4986 Note: By validating that requests are in the proper form, IPP objects force clients to use the proper form
4987 which, in turn, increases the chances that customers will be able to use such clients from multiple vendors
4988 with IPP objects from other vendors.

4989 16.3.4.3 Validate the presence of a single occurrence of required Operation attributes

4990 Client requests and IPP object responses contain Operation attributes that Section 3 requires to be
4991 present. Attributes within a group may be in any order, except for the ordering of target, charset, and
4992 natural languages attributes. These attributes must be first, and must be supplied in the following order:
4993 charset, natural language, and then target. An IPP object verifies that the attributes that Section 4
4994 requires to be supplied by the client have been supplied in the request (attributes without an * in the
4995 following tables). An asterisk (*) indicates groups and Operation attributes that the client may omit in a
4996 request or an IPP object may omit in a response.

4997 If an IPP object receives a request with required attributes missing or repeated from a group, the IPP
4998 object REJECTS the request and RETURNS the 'client-error-bad-request' status code. For example, it is
4999 an error for the "attributes-charset" or "attributes-natural-language" attribute to be omitted in any
5000 operation request, or for an Operation attribute to be supplied in a Job Template group or a Job Template
5001 attribute to be supplied in an Operation Attribute group in a create request. It is also an error to supply
5002 the "attributes-charset" attribute twice.

5003 Since these kinds of attribute errors are most likely to be detected by a client developer rather than by a
5004 customer, the IPP object NEED NOT return an indication of which attribute was in error in either the
5005 Unsupported Attributes group or the Status Message. Also, the IPP object NEED NOT find all attribute
5006 errors before returning this error.

5007 The following tables list all the attributes for all the operations by attribute group in each request and
5008 each response. The order of the groups is the order that the client supplies the groups as specified in
5009 Section 3. The order of the attributes within a group is arbitrary, except as noted for some of the special
5010 operation attributes (charset, natural language, and target). The tables below use the following notation:

5011 M indicates a MANDATORY attribute that an IPP object MUST support

5012 O indicates an OPTIONAL attribute that an IPP object NEED NOT support
5013 * indicates that a client MAY omit the attribute in a request and that an IPP object MAY
5014 omit the attribute in a response. The absence of an * means that a client MUST
5015 supply the attribute in a request and an IPP object MUST supply the attribute in a
5016 response.
5017
5018

Operation Requests

5019 The tables below show the attributes in their proper attribute groups for operation requests:

5020 Note: All operation requests contain the following common elements:
5021 version-number, operation-id, and request-id.
5022

Print-Job Request:

5023 Group 1: Operation Attributes (M)
5024 attributes-charset (M)
5025 attributes-natural-language (M)
5026 printer-uri (M)
5027 requesting-user-name (M*)
5028 job-name (M*)
5029 ipp-attribute-fidelity (M*)
5030 document-name (M*)
5031 document-format (M*)
5032 document-natural-language (O*)
5033 compression (O*)
5034 job-k-octets (O*)
5035 job-impressions (O*)
5036 job-media-sheets (O*)
5037 Group 2: Job Template Attributes (M)
5038 <Job Template attributes> (O*) (see Section 4.2)
5039 Group 3: Document Content (M)
5040 <document content>
5041
5042

Validate-Job Request:

5043 Group 1: Operation Attributes (M)
5044 attributes-charset (M)
5045 attributes-natural-language (M)
5046 printer-uri (M)
5047 requesting-user-name (M*)
5048 job-name (M*)
5049 ipp-attribute-fidelity (M*)
5050 document-name (M*)
5051 document-format (M*)
5052 document-natural-language (O*)
5053

5054 compression (O*)
5055 job-k-octets (O*)
5056 job-impressions (O*)
5057 job-media-sheets (O*)
5058 Group 2: Job Template Attributes (M)
5059 <Job Template attributes> (O*) (see Section 4.2)
5060
5061 Create-Job Request:
5062 Group 1: Operation Attributes (M)
5063 attributes-charset (M)
5064 attributes-natural-language (M)
5065 printer-uri (M)
5066 requesting-user-name (M*)
5067 job-name (M*)
5068 ipp-attribute-fidelity (M*)
5069 job-k-octets (O*)
5070 job-impressions (O*)
5071 job-media-sheets (O*)
5072 Group 2: Job Template Attributes (M)
5073 <Job Template attributes> (O*) (see Section 4.2)
5074
5075 Print-URI Request:
5076 Group 1: Operation Attributes (M)
5077 attributes-charset (M)
5078 attributes-natural-language (M)
5079 printer-uri (M)
5080 document-uri (M)
5081 requesting-user-name (M*)
5082 job-name (M*)
5083 ipp-attribute-fidelity (M*)
5084 document-name (M*)
5085 document-format (M*)
5086 document-natural-language (O*)
5087 compression (O*)
5088 job-k-octets (O*)
5089 job-impressions (O*)
5090 job-media-sheets (O*)
5091 Group 2: Job Template Attributes (M)
5092 <Job Template attributes> (O*) (see Section 4.2)
5093
5094 Send-Document Request:
5095 Group 1: Operation Attributes (M)
5096 attributes-charset (M)
5097 attributes-natural-language (M)
5098 (printer-uri & job-id) | job-uri (M)

5099 last-document (M)
5100 requesting-user-name (M*)
5101 document-name (M*)
5102 document-format (M*)
5103 document-natural-language (O*)
5104 compression (O*)
5105 Group 2: Document Content (M)
5106 <document content>
5107
5108 Send-URI Request:
5109 Group 1: Operation Attributes (M)
5110 attributes-charset (M)
5111 attributes-natural-language (M)
5112 (printer-uri & job-id) | job-uri (M)
5113 last-document (M)
5114 document-uri (M)
5115 requesting-user-name (M*)
5116 document-name (M*)
5117 document-format (M*)
5118 document-natural-language (O*)
5119 compression (O*)
5120
5121 Cancel-Job Request:
5122 Group 1: Operation Attributes (M)
5123 attributes-charset (M)
5124 attributes-natural-language (M)
5125 (printer-uri & job-id) | job-uri (M)
5126 requesting-user-name (M*)
5127 message (O*)
5128
5129 Get-Printer-Attributes Request:
5130 Group 1: Operation Attributes (M)
5131 attributes-charset (M)
5132 attributes-natural-language (M)
5133 printer-uri (M)
5134 requesting-user-name (M*)
5135 requested-attributes (M*)
5136 document-format (M*)
5137
5138 Get-Job-Attributes Request:
5139 Group 1: Operation Attributes (M)
5140 attributes-charset (M)
5141 attributes-natural-language (M)
5142 (printer-uri & job-id) | job-uri (M)
5143 requesting-user-name (M*)

5144 requested-attributes (M*)
5145
5146 Get-Jobs Request:
5147 Group 1: Operation Attributes (M)
5148 attributes-charset (M)
5149 attributes-natural-language (M)
5150 printer-uri (M)
5151 requesting-user-name (M*)
5152 limit (M*)
5153 requested-attributes (M*)
5154 which-jobs (M*)
5155 my-jobs (M*)
5156

5157 Operation Responses

5158 The tables below show the response attributes in their proper attribute groups for responses.

5159 Note: All operation responses contain the following common elements:
5160 version-number, status-code, and request-id.

5161
5162 Print-Job Response:
5163 Print-URI Response:
5164 Create-Job Response:
5165 Send-Document Response:
5166 Send-URI Response:
5167 Group 1: Operation Attributes (M)
5168 attributes-charset (M)
5169 attributes-natural-language (M)
5170 status-message (O*)
5171 Group 2: Unsupported Attributes (M*) (see Note 3)
5172 <unsupported attributes> (M*)
5173 Group 3: Job Object Attributes (M*) (see Note 2)
5174 job-uri (M)
5175 job-id (M)
5176 job-state (M)
5177 job-state-reasons (O*)
5178 job-state-message (O*)
5179 number-of-intervening-jobs (O*)
5180
5181 Validate-Job Response:
5182 Cancel-Job Response:
5183 Group 1: Operation Attributes (M)
5184 attributes-charset (M)

5185 attributes-natural-language (M)
5186 status-message (O*)
5187 Group 2: Unsupported Attributes (M*) (see Note 3)
5188 <unsupported attributes> (M*)
5189

5190 Note 2 - the Job Object Attributes and Printer Object Attributes are
5191 returned only if the IPP object returns one of the success status
5192 codes.

5193
5194 Note 3 - the Unsupported Attributes Group is present only if the
5195 client included some Operation and/or Job Template attributes that the
5196 Printer doesn't support whether a success or an error return.

5197
5198 Get-Printer-Attributes Response:
5199 Group 1: Operation Attributes (M)
5200 attributes-charset (M)
5201 attributes-natural-language (M)
5202 status-message (O*)
5203 Group 2: Unsupported Attributes (M*) (see Note 4)
5204 <unsupported attributes> (M*)
5205 Group 3: Printer Object Attributes(M*) (see Note 2)
5206 <requested attributes> (M*)
5207

5208 Note 4 - the Unsupported Attributes Group is present only if the
5209 client included some Operation attributes that the Printer doesn't
5210 support whether a success or an error return.

5211
5212 Get-Job-Attributes Response:
5213 Group 1: Operation Attributes (M)
5214 attributes-charset (M)
5215 attributes-natural-language (M)
5216 status-message (O*)
5217 Group 2: Unsupported Attributes (M*) (see Note 4)
5218 <unsupported attributes> (M*)
5219 Group 3: Job Object Attributes(M*) (see Note 2)
5220 <requested attributes> (M*)
5221

5222 Get-Jobs Response:
5223 Group 1: Operation Attributes (M)
5224 attributes-charset (M)
5225 attributes-natural-language (M)
5226 status-message (O*)
5227 Group 2: Unsupported Attributes (M*) (see Note 4)
5228 <unsupported attributes> (M*)
5229 Group 3: Job Object Attributes(M*) (see Note 2, 5)

5230 <requested attributes> (M*)

5231

5232 Note 5: for the Get-Jobs operation the response contains a separate
5233 Job Object Attributes group 3 to N containing requested-attributes for
5234 each job object in the response.
5235

5236 16.3.5 Validate the values of the MANDATORY Operation attributes

5237 An IPP object validates the values supplied by the client of the MANDATORY Operation attribute that
5238 the IPP object MUST support. The next section specifies the validation of the values of the OPTIONAL
5239 Operation attributes that IPP objects MAY support.

5240 The IPP object performs the following syntactic validation checks of each Operation attribute value:

5241

- 5242 a) that the length of each Operation attribute value is correct for the attribute syntax tag supplied
- 5243 by the client according to Section 4.1.
- 5244 b) that the attribute syntax tag is correct for that Operation attribute according to Section 3,
- 5245 c) that the value is in the range specified for that Operation attribute according to Section 3,
- 5246 d) that multiple values are supplied by the client only for operation attributes that are multi-
- 5247 valued, i.e., that are 1setOf X according to Section 3.

5248

5249 If any of these checks fail, the IPP object REJECTS the request and RETURNS the 'client-error-bad-
5250 request' or the 'client-error-request-value-too-long' status code. Since such an error is most likely to be
5251 an error detected by a client developer, rather than by an end-user, the IPP object NEED NOT return an
5252 indication of which attribute had the error in either the Unsupported Attributes Group or the Status
5253 Message. The description for each of these syntactic checks is explicitly expressed in the first IF
5254 statement in the following table.

5255 In addition, the IPP object checks each Operation attribute value against some Printer object attribute or
5256 some hard-coded value if there is no "xxx-supported" Printer object attribute defined. If its value is not
5257 among those supported or is not in the range supported, then the IPP object REJECTS the request and
5258 RETURNS the error status code indicated in the table by the second IF statement. If the value of the
5259 Printer object's "xxx-supported" attribute is 'no-value' (because the system administrator hasn't configured
5260 a value), the check always fails.

5261

attributes-charset (charset)

5262

5263 IF NOT any single non-empty 'charset' value less than or equal to 63 octets, REJECT/RETURN
5264 'client-error-request-value-too-long'.

5265

5266 IF NOT in the Printer object's "charset-supported" attribute, REJECT/RETURN "client-error-
charset-not-supported".

5267
5268 attributes-natural-language(naturalLanguage)
5269 IF NOT any single non-empty 'naturalLanguage' value less than or equal to 63 octets,
5270 REJECT/RETURN 'client-error-request-value-too-long'.
5271 ACCEPT the request even if not a member of the set in the Printer object's "generated-natural-
5272 language-supported" attribute.
5273
5274 requesting-user-name
5275 IF NOT any single 'name' value less than or equal to 255 octets, REJECT/RETURN 'client-error-
5276 request-value-too-long'.
5277 IF the IPP object can obtain a better authenticated name, use it instead.
5278
5279 job-name(name)
5280 IF NOT any single 'name' value less than or equal to 255 octets, REJECT/RETURN 'client-error-
5281 request-value-too-long'.
5282 IF NOT supplied by the client, the Printer object creates a name from the document-name or
5283 document-uri.
5284
5285 document-name (name)
5286 IF NOT any single 'name' value less than or equal to 255 octets, REJECT/RETURN 'client-error-
5287 request-value-too-long'.
5288
5289 ipp-attribute-fidelity (boolean)
5290 IF NOT either a single 'true' or 'false' 'boolean' value equal to 1 octet, REJECT/RETURN 'client-
5291 error-bad-request'.
5292 IF NOT supplied by the client, the IPP object assumes the value 'false'.
5293
5294 document-format (mimeMediaType)
5295 IF NOT any single non-empty 'mimeMediaType' value less than or equal to 255 octets,
5296 REJECT/RETURN 'client-error-request-value-too-long'.
5297 IF NOT in the Printer object's "document-format-supported" attribute, REJECT/RETURN 'client-
5298 error-document-format-not-supported'.
5299 IF NOT supplied by the client, the IPP object assumes the value of the Printer object's "document-
5300 format-default" attribute.
5301
5302 document-uri (uri)
5303 IF NOT any single non-empty 'uri' value less than or equal to 1023 octets, REJECT/RETURN 'client-
5304 error-request-value-too-long'.
5305 IF the URI syntax is not valid, REJECT/RETURN 'client-error-bad-request'.

5306 IF scheme is NOT in the Printer object's "reference-uri-schemes-supported" attribute,
5307 REJECT/RETURN 'client-error'-uri-scheme-not-supported'.
5308

5309 last-document (boolean)
5310 IF NOT either a single 'true' or 'false' 'boolean' value equal to 1 octet, REJECT/RETURN 'client-
5311 error-bad-request'.
5312

5313 job-id (integer(1:MAX))
5314 IF NOT any single 'integer' value equal to 4 octets AND in the range 1 to MAX, REJECT/RETURN
5315 'client-error-bad-request'.
5316 IF NOT a job-id of an existing Job object, REJECT/RETURN 'client-error-not-found' or 'client-error-
5317 gone' status code, if keep track of recently deleted jobs.
5318

5319 requested-attributes (1setOf keyword)
5320 IF NOT any number of 'keyword' values less than or equal to 255 octets, REJECT/RETURN 'client-
5321 error-request-value-too-long'.
5322 Ignore unsupported values which are the keyword names of unsupported attributes. Don't bother to
5323 copy such requested (unsupported) attributes to the Unsupported Attribute response group since
5324 the response will not return them.
5325

5326 which-jobs (type2 keyword)
5327 IF NOT a single 'keyword' value less than or equal to 255 octets, REJECT/RETURN 'client-error-
5328 request-value-too-long'.
5329 IF NEITHER 'completed' NOR 'not-completed', copy the attribute and the unsupported value to the
5330 Unsupported Attributes response group and REJECT/RETURN 'client-error-attributes-or-values-
5331 not-supported'.
5332 Note: a Printer still supports the 'completed' value even if it keeps no completed/canceled/aborted
5333 jobs: by returning no jobs when so queried.
5334 IF NOT supplied by the client, the IPP object assumes the 'not-completed' value.
5335

5336 my-jobs (boolean)
5337 IF NOT either a single 'true' or 'false' 'boolean' value equal to 1 octet, REJECT/RETURN 'client-
5338 error-bad-request'.
5339 IF NOT supplied by the client, the IPP object assumes the 'false' value.
5340

5341 limit (integer(1:MAX))
5342 IF NOT any single 'integer' value equal to 4 octets AND in the range 1 to MAX, REJECT/RETURN
5343 'client-error-bad-request'.
5344 IF NOT supplied by the client, the IPP object returns all jobs, no matter how many.
5345

5346 -----
5347

5348 16.3.6 Validate the values of the OPTIONAL Operation attributes

5349 OPTIONAL Operation attributes are those that an IPP object MAY or MAY NOT support. An IPP
5350 object validates the values of the OPTIONAL attributes supplied by the client. The IPP object performs
5351 the same syntactic validation checks for each OPTIONAL attribute value as in Section 16.3.5. As in
5352 Section 16.3.5, if any fail, the IPP object REJECTS the request and RETURNS the 'client-error-bad-
5353 request' or the 'client-error-request-value-too-long' status code.

5354 In addition, the IPP object checks each Operation attribute value against some Printer attribute or some
5355 hard-coded value if there is no "xxx-supported" Printer attribute defined. If its value is not among those
5356 supported or is not in the range supported, then the IPP object REJECTS the request and RETURNS the
5357 error status code indicated in the table. If the value of the Printer object's "xxx-supported" attribute is
5358 'no-value' (because the system administrator hasn't configured a value), the check always fails.

5359 If the IPP object doesn't recognize/support an attribute, the IPP object treats the attribute as an unknown
5360 or unsupported attribute (see the last row in the table below).

5361 -----
5362 document-natural-language (naturalLanguage)

5363 IF NOT any single non-empty 'naturalLanguage' value less than or equal to 63 octets,
5364 REJECT/RETURN 'client-error-request-value-too-long'.

5365 IF NOT a value that the Printer object supports in document formats, (no standard "xxx-supported"
5366 Printer attribute), REJECT/RETURN 'client-error-natural-language-not-supported'.

5367
5368 compression (type3 keyword)

5369 IF NOT any single 'keyword' values less than or equal to 255 octets, REJECT/RETURN 'client-error-
5370 request-value-too-long'.

5371 IF NOT in the Printer object's "compression-supported" attribute, copy the attribute and the
5372 unsupported value to the Unsupported Attributes response group and REJECT/RETURN 'client-
5373 error-attributes-or-values-not-supported'.

5374
5375 job-k-octets (integer(0:MAX))

5376 IF NOT any single 'integer' value equal to 4 octets,
5377 REJECT/RETURN 'client-error-bad-request'.

5378 IF NOT in the range of the Printer object's "job-k-octets-supported" attribute, copy the attribute and
5379 the unsupported value to the Unsupported Attributes response group and REJECT/RETURN
5380 'client-error-attributes-or-values-not-supported'.

5381

5382 job-impressions (integer(0:MAX))
5383 IF NOT any single 'integer' value equal to 4 octets,
5384 REJECT/RETURN 'client-error-bad-request'.
5385 IF NOT in the range of the Printer object's "job-impressions-supported" attribute, copy the attribute
5386 and the unsupported value to the Unsupported Attributes response group and REJECT/RETURN
5387 'client-error-attributes-or-values-not-supported'.
5388

5389 job-media-sheets (integer(0:MAX))
5390 IF NOT any single 'integer' value equal to 4 octets,
5391 REJECT/RETURN 'client-error-bad-request'.
5392 IF NOT in the range of the Printer object's "job-media-supported" attribute, copy the attribute and the
5393 unsupported value to the Unsupported Attributes response group and REJECT/RETURN 'client-
5394 error-attributes-or-values-not-supported'.
5395

5396 message (text(127))
5397 IF NOT any single 'text' value less than or equal to 127 octets,
5398 REJECT/RETURN 'client-error-request-value-too-long'.
5399

5400 unknown or unsupported attribute
5401 IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute
5402 syntax, REJECT/RETURN 'client-error-request-value-too-long'.
5403 ELSE copy the attribute and value to the Unsupported Attributes response group and change the
5404 attribute value to the "out-of-band" 'unsupported' value, but otherwise ignore the attribute.
5405

5406 Note: Future Operation attributes may be added to the protocol specification that may occur
5407 anywhere in the specified group. When the operation is otherwise successful, the IPP object returns
5408 the 'successful-ok-ignored-or-substituted-attributes' status code. Ignoring unsupported Operation
5409 attributes in all operations is analogous to the handling of unsupported Job Template attributes in the
5410 create and Validate-Job operations when the client supplies the "ipp-attribute-fidelity" Operation
5411 attribute with the 'false' value. This last rule is so that we can add OPTIONAL Operation attributes to
5412 future versions of IPP so that older clients can inter-work with new IPP objects and newer clients can
5413 inter-work with older IPP objects. (If the new attribute cannot be ignored without performing
5414 unexpectedly, the major version number would have been increased in the protocol document and in
5415 the request). This rule for Operation attributes is independent of the value of the "ipp-attribute-
5416 fidelity" attribute. For example, if an IPP object doesn't support the OPTIONAL "job-k-octets"
5417 attribute', the IPP object treats "job-k-octets" as an unknown attribute and only checks the length for
5418 the 'integer' attribute syntax supplied by the client. If it is not four octets, the IPP object REJECTS
5419 the request and RETURNS the 'client-error-bad-request' status code, else the IPP object copies the
5420 attribute to the Unsupported Attribute response group, setting the value to the "out-of-band"
5421 'unsupported' value, but otherwise ignores the attribute.

5422 -----

5423 16.4 Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add
5424 Documents

5425 This section in combination with the previous section recommends the processing steps for the Print-Job,
5426 Validate-Job, Print-URI, Create-Job, Send-Document, and Send-URI operations that IPP objects
5427 SHOULD use. These are the operations that create jobs, validate a Print-Job request, and add
5428 documents to a job.

5429 16.4.1 Default "ipp-attribute-fidelity" if not supplied

5430 The Printer object checks to see if the client supplied an "ipp-attribute-fidelity" Operation attribute. If the
5431 attribute is not supplied by the client, the IPP object assumes that the value is 'false'.

5432 16.4.2 Check that the Printer object is accepting jobs

5433 If the value of the Printer object's "printer-is-accepting-jobs" is 'false', the Printer object REJECTS the
5434 request and RETURNS the 'server-error-not-accepting-jobs' status code.

5435 16.4.3 Validate the values of the Job Template attributes

5436 An IPP object validates the values of all Job Template attribute supplied by the client. The IPP object
5437 performs the analogous syntactic validation checks of each Job Template attribute value that it performs
5438 for Operation attributes (see Section 16.3.5.):

- 5439 a) that the length of each value is correct for the attribute syntax tag supplied by the client
5440 according to Section 4.1.
5441 b) that the attribute syntax tag is correct for that attribute according to Sections 4.2 to 4.4,
5442 c) that multiple values are supplied only for multi-valued attributes, i.e., that are 1setOf X
5443 according to Sections 4.2 to 4.4
5444

5445 As in Section 16.3.5, if any of these syntactic checks fail, the IPP object REJECTS the request and
5446 RETURNS the 'client-error-bad-request' or 'client-error-request-value-too-long' status code, independent
5447 of the value of the "ipp-attribute-fidelity". Since such an error is most likely to be an error detected by a
5448 client developer, rather than by an end-user, the IPP object NEED NOT return an indication of which
5449 attribute had the error in either the Unsupported Attributes Group or the Status Message. The
5450 description for each of these syntactic checks is explicitly expressed in the first IF statement in the
5451 following table.

5452 In addition, the IPP object loops through all the client-supplied Job Template attributes, checking to see if
5453 the supplied attribute value(s) are supported or in the range supported, i.e., the value of the "xxx"
5454 attribute in the request is (1) a member of the set of values or is in the range of values of the Printer'
5455 objects "xxx-supported" attribute. If the value of the Printer object's "xxx-supported" attribute is 'no-
5456 value' (because the system administrator hasn't configured a value), the check always fails. If the check
5457 fails, the IPP object copies the attribute to the Unsupported Attributes response group with its
5458 unsupported value. If the attribute contains more than one value, each value is checked and each
5459 unsupported value is separately copied, while supported values are not copied. If an IPP object doesn't
5460 recognize/support a Job Template attribute, i.e., there is no corresponding Printer object "xxx-supported"
5461 attribute, the IPP object treats the attribute as an unknown or unsupported attribute (see the last row in
5462 the table below).

5463 If some Job Template attributes are supported for some document formats and not for others or the
5464 values are different for different document formats, the IPP object SHOULD take that into account in
5465 this validation using the value of the "document-format" supplied by the client (or defaulted to the value
5466 of the Printer's "document-format-default" attribute, if not supplied by the client). For example, if
5467 "number-up" is supported for the 'text/plain' document format, but not for the 'application/postscript'
5468 document format, the check SHOULD (though it NEED NOT) depend on the value of the "document-
5469 format" operation attribute. See "document-format" in section 3.2.1.1 and 3.2.5.1.

5470 Note: whether the request is accepted or rejected is determined by the value of the "ipp-attribute-fidelity"
5471 attribute in a subsequent step, so that all Job Template attribute supplied are examined and all
5472 unsupported attributes and/or values are copied to the Unsupported Attributes response group.

5473 -----

5474 job-priority (integer(1:100))

5475 IF NOT any single 'integer' value equal to 4 octets, REJECT/RETURN 'client-error-bad-request'.

5476 IF NOT supplied by the client, use the value of the Printer object's "job-priority-default" attribute at
5477 job submission time.

5478 IF NOT in the range 1 to 100, inclusive, copy the attribute and the unsupported value to the
5479 Unsupported Attributes response group.

5480 Map the value to the nearest supported value in the range 1:100 as specified by the number of
5481 discrete values indicated by the value of the Printer's "job-priority-supported" attribute. See the
5482 formula in Section 4.2.1.

5483

5484 job-hold-until (type3 keyword | name)

5485 IF NOT any single 'keyword' or 'name' value less than or equal to 255 octets, REJECT/RETURN
5486 'client-error-request-value-too-long'.

5487 IF NOT supplied by the client, use the value of the Printer object's "job-hold-until" attribute at job
5488 submission time.

5489 IF NOT in the Printer object's "job-hold-until-supported" attribute, copy the attribute and the
5490 unsupported value to the Unsupported Attributes response group.
5491
5492 job-sheets (type3 keyword | name)
5493 IF NOT any single 'keyword' or 'name' value less than or equal to 255 octets, REJECT/RETURN
5494 'client-error-request-value-too-long'.
5495 IF NOT in the Printer object's "job-sheets-supported" attribute, copy the attribute and the
5496 unsupported value to the Unsupported Attributes response group.
5497
5498 multiple-document-handling (type2 keyword)
5499 IF NOT any single 'keyword' value less than or equal to 255 octets, REJECT/RETURN 'client-error-
5500 request-value-too-long'.
5501 IF NOT in the Printer object's "multiple-document-handling-supported" attribute, copy the attribute
5502 and the unsupported value to the Unsupported Attributes response group.
5503
5504 copies (integer(1:MAX))
5505 IF NOT any single 'integer' value equal to 4 octets,
5506 REJECT/RETURN 'client-error-bad-request'.
5507 IF NOT in range of the Printer object's "copies-supported" attribute
5508 copy the attribute and the unsupported value to the Unsupported Attributes response group.
5509
5510 finishings (1setOf type2 enum)
5511 IF NOT any 'enum' value(s) equal to 4 octets, REJECT/RETURN 'client-error-bad-request'.
5512 IF NOT in the Printer object's "finishings-supported" attribute, copy the attribute and the unsupported
5513 value(s), but not any supported values, to the Unsupported Attributes response group.
5514
5515 page-ranges (1setOf rangeOfInteger(1:MAX))
5516 IF NOT any 'rangeOfInteger' value(s) each equal to 8 octets, REJECT/RETURN 'client-error-bad-
5517 request'.
5518 IF first value is greater than second value in any range, the ranges are not in ascending order, or
5519 ranges overlap, REJECT/RETURN 'client-error-bad-request'.
5520 IF the value of the Printer object's "page-ranges-supported" attribute is 'false', copy the attribute to
5521 the Unsupported Attributes response group and set the value to the "out-of-band" 'unsupported'
5522 value.
5523
5524 sides (type2 keyword)
5525 IF NOT any single 'keyword' value less than or equal to 255 octets, REJECT/RETURN 'client-error-
5526 request-value-too-long'.
5527 IF NOT in the Printer object's "sides-supported" attribute, copy the attribute and the unsupported
5528 value to the Unsupported Attributes response group.

5529
5530 number-up (integer(1:MAX))
5531 IF NOT any single 'integer' value equal to 4 octets,
5532 REJECT/RETURN 'client-error-bad-request'.
5533 IF NOT a value or in the range of one of the values of the Printer object's "number-up-supported"
5534 attribute, copy the attribute and value to the Unsupported Attribute response group.
5535
5536 orientation-requested (type2 enum)
5537 IF NOT any single 'enum' value equal to 4 octets,
5538 REJECT/RETURN 'client-error-bad-request'.
5539 IF NOT in the Printer object's "orientation-requested-supported" attribute, copy the attribute and the
5540 unsupported value to the Unsupported Attributes response group.
5541
5542 media (type3 keyword | name)
5543 IF NOT any single 'keyword' or 'name' value less than or equal to 255 octets, REJECT/RETURN
5544 'client-error-request-value-too-long'.
5545 IF NOT in the Printer object's "media-supported" attribute, copy the attribute and the unsupported
5546 value to the Unsupported Attributes response group.
5547
5548 printer-resolution (resolution)
5549 IF NOT any single 'resolution' value equal to 9 octets,
5550 REJECT/RETURN 'client-error-bad-request'.
5551 IF NOT in the Printer object's "multiple-document-handling-supported" attribute, copy the attribute
5552 and the unsupported value to the Unsupported Attributes response group.
5553
5554 print-quality (type2 enum)
5555 IF NOT any single 'enum' value equal to 4 octets,
5556 REJECT/RETURN 'client-error-bad-request'.
5557 IF NOT in the Printer object's "print-quality-supported" attribute, copy the attribute and the
5558 unsupported value to the Unsupported Attributes response group.
5559
5560 unknown or unsupported attribute (i.e., there is no corresponding Printer object "xxx-supported"
5561 attribute)
5562 IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute
5563 syntax,
5564 REJECT/RETURN 'client-error-bad-request' or 'client-error-request-value-too-long'.
5565 ELSE copy the attribute and value to the Unsupported Attributes response group and change the
5566 attribute value to the "out-of-band" 'unsupported' value. Any remaining Job Template Attributes
5567 are either unknown or unsupported Job Template attributes and are validated algorithmically
5568 according to their attribute syntax for proper length (see below).

5569 -----

5570
 5571 If the attribute syntax is supported AND the length check fails, the IPP object REJECTS the request and
 5572 RETURNS the 'client-error-request-value-too-long' status code, else the IPP object copies the
 5573 unsupported Job Template attribute to the Unsupported Attributes response group and changes the
 5574 attribute value to the "out-of-band" 'unsupported' value. The following table shows the length checks for
 5575 all attribute syntaxes. In the following table: "<=" means less than or equal, "=" means equal to:

5576 Name	5576 Octet length check for read-write attributes	
5577 -----	5577 -----	
5578 'textWithLanguage	<= 1023 AND	'naturalLanguage' <= 63
5579 'textWithoutLanguage'	<= 1023	
5580 'nameWithLanguage'	<= 255 AND	'naturalLanguage' <= 63
5581 'nameWithoutLanguage'	<= 255	
5582 'keyword'	<= 255	
5583 'enum'	= 4	
5584 'uri'	<= 1023	
5585 'uriScheme'	<= 63	
5586 'charset'	<= 63	
5587 'naturalLanguage'	<= 63	
5588 'mimeType'	<= 255	
5589 'octetString'	<= 1023	
5590 'boolean'	= 1	
5591 'integer'	= 4	
5592 'rangeOfInteger'	= 8	
5593 'dateTime'	= 11	
5594 'resolution'	= 9	
5595 'lsetOf X'		
5596		

5597 16.4.4 Check for conflicting Job Template attributes values

5598 Once all the Operation and Job Template attributes have been checked individually, the Printer object
 5599 SHOULD check for any conflicting values among all the supported values supplied by the client. For
 5600 example, a Printer object might be able to staple and to print on transparencies, however due to physical
 5601 stapling constraints, the Printer object might not be able to staple transparencies. The IPP object copies
 5602 the supported attributes and their conflicting attribute values to the Unsupported Attributes response
 5603 group. The Printer object only copies over those attributes that the Printer object either ignores or
 5604 substitutes in order to resolve the conflict, and it returns the original values which were supplied by the
 5605 client. For example suppose the client supplies "finishings" equals 'staple' and "media" equals
 5606 'transparency', but the Printer object does not support stapling transparencies. If the Printer chooses to
 5607 ignore the stapling request in order to resolve the conflict, the Printer objects returns "finishings" equal to

5608 'staple' in the Unsupported Attributes response group. If any attributes are multi-valued, only the
5609 conflicting values of the attributes are copied.

5610 Note: The decisions made to resolve the conflict (if there is a choice) is implementation dependent.

5611 16.4.5 Decide whether to REJECT the request

5612 If there were any unsupported Job Template attributes or unsupported/conflicting Job Template attribute
5613 values and the client supplied the "ipp-attribute-fidelity" attribute with the 'true' value, the Printer object
5614 REJECTS the request and return the status code:

5615 (1) 'client-error-conflicting-attributes' status code, if there were any conflicts between attributes
5616 supplied by the client.

5617 (2) 'client-error-attributes-or-values-not-supported' status code, otherwise.
5618

5619 Note: Unsupported Operation attributes or values that are returned do not affect the status returned in
5620 this step. If the unsupported Operation attribute was a serious error, the above already rejected the
5621 request in a previous step. If control gets to this step with unsupported Operation attributes being
5622 returned, they are not serious errors.

5623 16.4.6 For the Validate-Job operation, RETURN one of the success status codes

5624 If the requested operation is the Validate-Job operation, the Printer object returns:

5625 (1) the "successful-ok" status code, if there are no unsupported or conflicting Job Template attributes
5626 or values.

5627 (2) the "successful-ok-conflicting-attributes, if there are any conflicting Job Template attribute or
5628 values.

5629 (3) the "successful-ok-ignored-or-substituted-attributes, if there are only unsupported Job Template
5630 attributes or values.
5631

5632 Note: Unsupported Operation attributes or values that are returned do not affect the status returned in
5633 this step. If the unsupported Operation attribute was a serious error, the above already rejected the
5634 request in a previous step. If control gets to this step with unsupported Operation attributes being
5635 returned, they are not serious errors.

5636 16.4.7 Create the Job object with attributes to support

5637 If "ipp-attribute-fidelity" is set to 'false' (or it was not supplied by the client), the Printer object:

- 5638 (1) creates a Job object, assigns a unique value to the job's "job-uri" and "job-id" attributes, and
5639 initializes all of the job's other supported Job Description attributes.
- 5640 (2) removes all unsupported attributes from the Job object.
- 5641 (3) for each unsupported value, removes either the unsupported value or substitutes the unsupported
5642 attribute value with some supported value. If an attribute has no values after removing
5643 unsupported values from it, the attribute is removed from the Job object (so that the normal
5644 default behavior at job processing time will take place for that attribute).
- 5645 (4) for each conflicting value, removes either the conflicting value or substitutes the conflicting
5646 attribute value with some other supported value. If an attribute has no values after removing
5647 conflicting values from it, the attribute is removed from the Job object (so that the normal default
5648 behavior at job processing time will take place for that attribute).

5649

5650 If there were no attributes or values flagged as unsupported, or the value of "ipp-attribute-fidelity" was
5651 'false', the Printer object is able to accept the create request and create a new Job object. If the "ipp-
5652 attribute-fidelity" attribute is set to 'true', the Job Template attributes that populate the new Job object are
5653 necessarily all the Job Template attributes supplied in the create request. If the "ipp-attribute-fidelity"
5654 attribute is set to 'false', the Job Template attributes that populate the new Job object are all the client
5655 supplied Job Template attributes that are supported or that have value substitution. Thus, some of the
5656 requested Job Template attributes may not appear in the Job object because the Printer object did not
5657 support those attributes. The attributes that populate the Job object are persistently stored with the Job
5658 object for that Job. A Get-Job-Attributes operation on that Job object will return only those attributes
5659 that are persistently stored with the Job object.

5660 Note: All Job Template attributes that are persistently stored with the Job object are intended to be
5661 "override values"; that is, they that take precedence over whatever other embedded instructions might be
5662 in the document data itself. However, it is not possible for all Printer objects to realize the semantics of
5663 "override". End users may query the Printer's "pdl-override-supported" attribute to determine if the
5664 Printer either attempts or does not attempt to override document data instructions with IPP attributes.

5665 There are some cases, where a Printer supports a Job Template attribute and has an associated default
5666 value set for that attribute. In the case where a client does not supply the corresponding attribute, the
5667 Printer does not use its default values to populate Job attributes when creating the new Job object; only
5668 Job Template attributes actually in the create request are used to populate the Job object. The Printer's
5669 default values are only used later at Job processing time if no other IPP attribute or instruction embedded
5670 in the document data is present.

5671 Note: If the default values associated with Job Template attributes that the client did not supply were to
5672 be used to populate the Job object, then these values would become "override values" rather than
5673 defaults. If the Printer supports the 'attempted' value of the "pdl-override-supported" attribute, then these
5674 override values could replace values specified within the document data. This is not the intent of the
5675 default value mechanism. A default value for an attribute is used only if the create request did not specify

5676 that attribute (or it was ignored when allowed by "ipp-attribute-fidelity" being 'false') and no value was
5677 provided within the content of the document data.

5678 If the client does not supply a value for some Job Template attribute, and the Printer does not support
5679 that attribute, as far as IPP is concerned, the result of processing that Job (with respect to the missing
5680 attribute) is undefined.

5681 16.4.8 Return one of the success status codes

5682 Once the Job object has been created, the Printer object accepts the request and returns to the client:

- 5683 (1) the 'successful-ok' status code, if there are no unsupported or conflicting Job Template attributes
5684 or values.
- 5685 (2) the 'successful-ok-conflicting-attributes' status code, if there are any conflicting Job Template
5686 attribute or values.
- 5687 (3) the 'successful-ok-ignored-or-substituted-attributes' status code, if there are only unsupported Job
5688 Template attributes or values.

5689

5690 Note: Unsupported Operation attributes or values that are returned do not affect the status returned in
5691 this step. If the unsupported Operation attribute was a serious error, the above already rejected the
5692 request in a previous step. If control gets to this step with unsupported Operation attributes being
5693 returned, they are not serious errors.

5694 The Printer object also returns Job status attributes that indicate the initial state of the Job ('pending',
5695 'pending-held', 'processing', etc.), etc. See Print-Job Response, section 3.2.1.2.

5696 16.4.9 Accept appended Document Content

5697 The Printer object accepts the appended Document Content data and either starts it printing, or spools it
5698 for later processing.

5699 16.4.10 Scheduling and Starting to Process the Job

5700 The Printer object uses its own configuration and implementation specific algorithms for scheduling the
5701 Job in the correct processing order. Once the Printer object begins processing the Job, the Printer
5702 changes the Job's state to 'processing'. If the Printer object supports PDL override (the "pdl-override-
5703 supported" attribute set to 'attempted'), the implementation does its best to see that IPP attributes take
5704 precedence over embedded instructions in the document data.

5705 16.4.11 Completing the Job

5706 The Printer object continues to process the Job until it can move the Job into the 'completed' state. If an
5707 Cancel-Job operation is received, the implementation eventually moves the Job into the 'canceled' state.
5708 If the system encounters errors during processing that do not allow it to progress the Job into a
5709 completed state, the implementation halts all processing, cleans up any resources, and moves the Job into
5710 the 'aborted' state.

5711 16.4.12 Destroying the Job after completion

5712 Once the Job moves to the 'completed', 'aborted', or 'canceled' state, it is an implementation decision as to
5713 when to destroy the Job object and release all associated resources. Once the Job has been destroyed, the
5714 Printer would return either the "client-error-not-found" or "client-error-gone" status codes for operations
5715 directed at that Job.

5716 Note: the Printer object SHOULD NOT re-use a "job-uri" or "job-id" value for a sufficiently long time
5717 after a job has been destroyed, so that stale references kept by clients are less likely to access the wrong
5718 (newer) job.

5719 16.4.13 Interaction with "ipp-attribute-fidelity"

5720 Some Printer object implementations may support "ipp-attribute-fidelity" set to 'true' and "pdl-override-
5721 supported" set to 'attempted' and yet still not be able to realize exactly what the client specifies in the
5722 create request. This is due to legacy decisions and assumptions that have been made about the role of job
5723 instructions embedded within the document data and external job instructions that accompany the
5724 document data and how to handle conflicts between such instructions. The inability to be 100% precise
5725 about how a given implementation will behave is also compounded by the fact that the two special
5726 attributes, "ipp-attribute-fidelity" and "pdl-override-supported", apply to the whole job rather than
5727 specific values for each attribute. For example, some implementations may be able to override almost all
5728 Job Template attributes except for "number-up".

5729 16.5 Using Job Template Attributes During Document Processing.

5730 The Printer object uses some of the Job object's Job Template attributes during the processing of the
5731 document data associated with that job. These include, but are not limited to, "orientation", "number-
5732 up", "sides", "media", and "copies". The processing of each document in a Job Object SHALL follow the
5733 steps below. These steps are intended only to identify when and how attributes are to be used in
5734 processing document data and any alternative steps that accomplishes the same effect can be used to
5735 implement this specification.

- 5736 1. Using the client supplied "document-format" attribute or some form of document format detection
5737 algorithm (if the value of "document-format" is not specific enough), determine whether or not
5738 the document data has already been formatted for printing. If the document data has been
5739 formatted, then go to step 2. Otherwise, the document data SHALL be formatted. The formatting
5740 detection algorithm is implementation defined and is not specified by this specification. The
5741 formatting of the document data uses the "orientation-requested" attribute to determine how the
5742 formatted print data should be placed on a print-stream page, see section 4.2.10 for the details.
5743
- 5744 2. The document data is in the form of a print-stream in a known media type. The "page-ranges"
5745 attribute is used to select, as specified in section 4.2.7, a sub-sequence of the pages in the print-
5746 stream that are to be processed and images.
5747
- 5748 3. The input to this step is a sequence of print-stream pages. This step is controlled by the "number-
5749 up" attribute. If the value of "number-up" is N, then during the processing of the print-stream
5750 pages, each N print-stream pages are positioned, as specified in section 4.2.9, to create a single
5751 impression. If a given document does not have N more print-stream pages, then the completion of
5752 the impression is controlled by the "multiple-document-handling" attribute as described in section
5753 4.2.4; when the value of this attribute is 'single-document', the print-stream pages of document
5754 data from subsequent documents is used to complete the impression.
5755
- 5756 The size(scaling), position(translation) and rotation of the print-stream pages on the impression is
5757 implementation defined. Note that during this process the print-stream pages may be rendered to
5758 a form suitable for placing on the impression; this rendering is controlled by the values of the
5759 "printer-resolution" and "print-quality" attributes as described in sections 4.2.12 and 4.2.13. In the
5760 case N=1, the impression is nearly the same as the print-stream page; the differences would only
5761 be in the size, position and rotation of the print-stream page and/or any decoration, such as a
5762 frame to the page, that is added by the implementation.
5763
- 5764 4. The collection of impressions is placed, in sequence, onto sides of the media sheets. This placement
5765 is controlled by the "sides" attribute and the orientation of the print-stream page, as described in
5766 section 4.2.8. The orientation of the print-stream pages affects the orientation of the impression;
5767 for example, if "number-up" equals 2, then, typically, two portrait print-stream pages become one
5768 landscape impression. Note that the placement of impressions onto media sheets is also controlled
5769 by the "multiple-document-handling" attribute as described in section 4.2.4.
5770
- 5771 5. The "copies" and "multiple-document-handling" attributes are used to determine how many copies
5772 of each media instance are created and in what order. See sections 4.2.5 and 4.2.4 for the details.
5773
- 5774 6. When the correct number of copies are created, the media instances are finished according to the
5775 values of the "finishings" attribute as described in 4.2.6. Note that sometimes finishing operations

5776 may require manual intervention to perform the finishing operations on the copies, especially
 5777 uncollated copies. This specification allows any or all of the processing steps to be performed
 5778 automatically or manually at the discretion of the Printer object.

5779 17. APPENDIX E: Generic Directory Schema

5780 This section defines a generic schema for an entry in a directory service. A directory service is a means
 5781 by which service users can locate service providers. In IPP environments, this means that IPP Printers
 5782 can be registered (either automatically or with the help of an administrator) as entries of type printer in
 5783 the directory using an implementation specific mechanism such as entry attributes, entry type fields,
 5784 specific branches, etc. IPP clients can search or browse for entries of type printer. Clients use the
 5785 directory service to find entries based on naming, organizational contexts, or filtered searches on attribute
 5786 values of entries. For example, a client can find all printers in the "Local Department" context.
 5787 Authentication and authorization are also often part of a directory service so that an administrator can
 5788 place limits on end users so that they are only allowed to find entries to which they have certain access
 5789 rights. IPP itself does not require any specific directory service protocol or provider.

5790 Note: Some directory implementations allow for the notion of "aliasing". That is, one directory entry
 5791 object can appear as multiple directory entry object with different names for each object. In each case,
 5792 each alias refers to the same directory entry object which refers to a single IPP Printer object.

5793 The generic schema is a subset of IPP Printer Job Template and Printer Description attributes (sections
 5794 4.2 and 4.4). These attributes are identified as either MANDATORY or OPTIONAL for the directory
 5795 entry itself. This conformance labeling is NOT the same conformance labeling applied to the attributes of
 5796 IPP Printers objects. MANDATORY attributes MUST be associated with each directory entry.
 5797 OPTIONAL attributes SHOULD be associated with the directory entry (if known or supported). In
 5798 addition, all directory entry attributes SHOULD reflect the current attribute values for the corresponding
 5799 Printer object.

5800 In order to bridge between the directory service and the IPP Printer object, one of the MANDATORY
 5801 directory entry attributes is the Printer object's "printer-uri-supported" attribute. The IPP client queries
 5802 the "printer-uri-supported" attribute in the directory entry and then addresses the IPP Printer object using
 5803 one of its URIs. The "uri-security-supported" attribute identifies the protocol (if any) used to secure a
 5804 channel.

5805 The following attributes define the generic schema for directory entries of type PRINTER:

5806	printer-uri-supported	MANDATORY	Section 4.4.1
5807	uri-security-supported	MANDATORY	Section 4.4.2
5808	printer-name	MANDATORY	Section 4.4.3

5809	printer-location	OPTIONAL	Section 4.4.4
5810	printer-info	OPTIONAL	Section 4.4.5
5811	printer-more-info	OPTIONAL	Section 4.4.6
5812	printer-make-and-model	OPTIONAL	Section 4.4.8
5813	charset-supported	MANDATORY	Section 4.4.15
5814	generated-natural-language-		
5815	supported	MANDATORY	Section 4.4.17
5816	document-format-supported	OPTIONAL	Section 4.4.19
5817	color-supported	OPTIONAL	Section 4.4.23
5818	finishings-supported	OPTIONAL	Section 4.2.6
5819	number-up-supported	OPTIONAL	Section 4.2.7
5820	sides-supported	OPTIONAL	Section 4.2.8
5821	media-supported	OPTIONAL	Section 4.2.11
5822	printer-resolution-supported	OPTIONAL	Section 4.2.12
5823	print-quality-supported	OPTIONAL	Section 4.2.13
5824			