

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~~ Astart Technologies

June ~~1930~~, 1998

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

28 Abstract

29 This document is one of a set of documents, which together describe all aspects of a new Internet  
30 Printing Protocol (IPP). IPP is an application level protocol that can be used for distributed printing  
31 using Internet tools and technologies. The protocol is heavily influenced by the printing model  
32 introduced in the Document Printing Application (DPA) [ISO10175] standard. Although DPA specifies  
33 both end user and administrative features, IPP version 1.0 (IPP/1.0) focuses only on end user  
34 functionality.

35 The full set of IPP documents includes:

- 36 Design Goals for an Internet Printing Protocol [IPP-REQ] (informational)
- 37 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [IPP-RAT]
- 38 (informational)
- 39 Internet Printing Protocol/1.0: Model and Semantics (this document)
- 40 Internet Printing Protocol/1.0: Encoding and Transport [IPP-PRO]
- 41 Mapping between LPD and IPP Protocols [IPP LPD] (informational)

42

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

60 ~~This document is one of a set of documents, which together describe all aspects of a new Internet~~  
61 ~~Printing Protocol (IPP). IPP is an application level protocol that can be used for distributed printing~~  
62 ~~using Internet tools and technologies. The protocol is heavily influenced by the printing model~~  
63 ~~introduced in the Document Printing Application (DPA) [ISO10175] standard. Although DPA specifies~~  
64 ~~both end-user and administrative features, IPP version 1.0 (IPP/1.0) focuses only on end-user~~  
65 ~~functionality.~~

66 ~~The full set of IPP documents includes:~~

- 67 ~~Requirements for an Internet Printing Protocol [IPP-REQ]~~
- 68 ~~Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [IPP-RAT]~~
- 69 ~~Internet Printing Protocol/1.0: Model and Semantics (this document)~~
- 70 ~~Internet Printing Protocol/1.0: Transport and Encoding [IPP-PRO]~~

71

72 ~~The requirements document, "Requirements for an Internet Printing Protocol", takes a broad look at~~  
73 ~~distributed printing functionality, and it enumerates real life scenarios that help to clarify the features that~~  
74 ~~need to be included in a printing protocol for the Internet. It identifies requirements for three types of~~  
75 ~~users: end users, operators, and administrators. The requirements document calls out a subset of end~~  
76 ~~user requirements that are satisfied in IPP/1.0. Operator and administrator requirements are out of scope~~  
77 ~~for version 1.0. The rationale document, "Rationale for the Structure and Model and Protocol for the~~  
78 ~~Internet Printing Protocol", describes IPP from a high level view, defines a roadmap for the various~~  
79 ~~documents that form the suite of IPP specifications, and gives background and rationale for the IETF~~  
80 ~~working group's major decisions. This document, "Internet Printing Protocol/1.0: Model and Semantics",~~  
81 ~~describes a simplified model with abstract objects, their attributes, and their operations. The model~~  
82 ~~introduces a Printer and a Job. The Job supports multiple documents per Job. The model document also~~  
83 ~~addresses how security, internationalization, and directory issues are addressed. The protocol~~  
84 ~~specification, " Internet Printing Protocol/1.0: Transport and Encoding", is a formal mapping of the~~  
85 ~~abstract operations and attributes defined in the model document onto HTTP/1.1. The protocol~~  
86 ~~specification defines the encoding rules for a new Internet media type called "application/ipp".~~

87	Table of Contents	
88	1. Introduction.....	10
89	1.1 Simplified Printing Model.....	12
90	2. IPP Objects .....	14
91	2.1 Printer Object.....	14
92	2.2 Job Object.....	16
93	2.3 Object Relationships.....	17
94	2.4 Object Identity .....	18
95	3. IPP Operations .....	20
96	3.1 Common Semantics .....	21
97	Required Parameters.....	21
98	3.1.2 Operation IDs and Request IDs .....	22
99	3.1.3 Attributes .....	22
100	3.1.4 Character Set and Natural Language Operation Attributes.....	24
101	3.1.4.1 Request Operation Attributes.....	24
102	3.1.4.2 Response Operation Attributes .....	27
103	3.1.5 Operation Targets .....	28
104	3.1.6 Operation Status Codes and Messages.....	30
105	3.1.7 Versions.....	31
106	3.1.8 Job Creation Operations .....	32
107	3.2 Printer Operations.....	34
108	3.2.1 Print-Job Operation.....	34
109	3.2.1.1 Print-Job Request .....	34
110	3.2.1.2 Print-Job Response .....	38
111	3.2.2 Print-URI Operation.....	41
112	3.2.3 Validate-Job Operation.....	41
113	3.2.4 Create-Job Operation .....	42
114	3.2.5 Get-Printer-Attributes Operation.....	42
115	3.2.5.1 Get-Printer-Attributes Request .....	42
116	3.2.5.2 Get-Printer-Attributes Response .....	44
117	3.2.6 Get-Jobs Operation .....	45
118	3.2.6.1 Get-Jobs Request .....	45
119	3.2.6.2 Get-Jobs Response .....	47
120	3.3 Job Operations.....	48
121	3.3.1 Send-Document Operation .....	48
122	3.3.1.1 Send-Document Request .....	49
123	3.3.1.2 Send-Document Response .....	51
124	3.3.2 Send-URI Operation .....	52
125	3.3.3 Cancel-Job Operation.....	52

126	3.3.3.1	Cancel-Job Request .....	52
127	3.3.3.2	Cancel-Job Response .....	53
128	3.3.4	Get-Job-Attributes Operation .....	53
129	3.3.4.1	Get-Job-Attributes Request .....	54
130	3.3.4.2	Get-Job-Attributes Response .....	55
131	4.	Object Attributes .....	56
132	4.1	Attribute Syntaxes.....	56
133	4.1.1	'text' .....	57
134	4.1.1.1	'textWithoutLanguage' .....	58
135	4.1.1.2	'textWithLanguage' .....	58
136	4.1.2	'name' .....	59
137	4.1.2.1	'nameWithoutLanguage' .....	59
138	4.1.2.2	'nameWithLanguage' .....	59
139	4.1.3	'keyword' .....	60
140	4.1.4	'enum' .....	61
141	4.1.5	'uri' .....	62
142	4.1.6	'uriScheme' .....	62
143	4.1.7	'charset' .....	62
144	4.1.8	'naturalLanguage' .....	63
145	4.1.9	'mimeMediaType' .....	63
146	4.1.10	'octetString' .....	65
147	4.1.11	'boolean' .....	65
148	4.1.12	'integer' .....	65
149	4.1.13	'rangeOfInteger' .....	65
150	4.1.14	'dateTime' .....	65
151	4.1.15	'resolution' .....	65
152	4.1.16	'1setOf X' .....	66
153	4.2	Job Template Attributes .....	66
154	4.2.1	job-priority (integer(1:100)) .....	70
155	4.2.2	job-hold-until (type3 keyword   name (MAX)).....	71
156	4.2.3	job-sheets (type3 keyword   name(MAX)) .....	72
157	4.2.4	multiple-document-handling (type2 keyword).....	72
158	4.2.5	copies (integer(1:MAX)) .....	73
159	4.2.6	finishings (1setOf type2 enum) .....	73
160	4.2.7	page-ranges (1setOf rangeOfInteger (1:MAX)) .....	74
161	4.2.8	sides (type2 keyword) .....	75
162	4.2.9	number-up (integer(1:MAX)) .....	76
163	4.2.10	orientation-requested (type2 enum) .....	76
164	4.2.11	media (type3 keyword   name(MAX)) .....	77
165	4.2.12	printer-resolution (resolution).....	78

166	4.2.13	print-quality (type2 enum) .....	78
167	4.3	Job Description Attributes.....	78
168	4.3.1	job-uri (uri) .....	80
169	4.3.2	job-id (integer(1:MAX)).....	80
170	4.3.3	job-printer-uri (uri).....	81
171	4.3.4	job-more-info (uri) .....	81
172	4.3.5	job-name (name(MAX)).....	81
173	4.3.6	job-originating-user-name (name(MAX)) .....	81
174	4.3.7	job-state (type1 enum).....	82
175	4.3.8	job-state-reasons (1setOf type2 keyword).....	84
176	4.3.9	job-state-message (text(MAX)).....	86
177	4.3.10	number-of-documents (integer(0:MAX)).....	87
178	4.3.11	output-device-assigned (name(127)).....	87
179	4.3.12	time-at-creation (integer(0:MAX)) .....	87
180	4.3.13	time-at-processing (integer(0:MAX)) .....	87
181	4.3.14	time-at-completed (integer(0:MAX)).....	87
182	4.3.15	number-of-intervening-jobs (integer(0:MAX)).....	88
183	4.3.16	job-message-from-operator (text(127)).....	88
184	4.3.17	job-k-octets (integer(0:MAX)) .....	88
185	4.3.18	job-impressions (integer(0:MAX)).....	88
186	4.3.19	job-media-sheets (integer(0:MAX)).....	89
187	4.3.20	job-k-octets-processed (integer(0:MAX)).....	89
188	4.3.21	job-impressions-completed (integer(0:MAX)).....	90
189	4.3.22	job-media-sheets-completed (integer(0:MAX)).....	90
190	4.3.23	attributes-charset (charset) .....	90
191	4.3.24	attributes-natural-language (naturalLanguage).....	90
192	4.4	Printer Description Attributes.....	91
193	4.4.1	printer-uri-supported (1setOf uri) .....	94
194	4.4.2	uri-security-supported (1setOf type2 keyword).....	94
195	4.4.3	printer-name (name(127)).....	95
196	4.4.4	printer-location (text(127)).....	95
197	4.4.5	printer-info (text(127)) .....	95
198	4.4.6	printer-more-info (uri) .....	95
199	4.4.7	printer-driver-installer (uri).....	96
200	4.4.8	printer-make-and-model (text(127)) .....	96
201	4.4.9	printer-more-info-manufacturer (uri) .....	96
202	4.4.10	printer-state (type1 enum) .....	96
203	4.4.11	printer-state-reasons (1setOf type2 keyword) .....	97
204	4.4.12	printer-state-message (text(MAX)).....	100
205	4.4.13	operations-supported (1setOf type2 enum).....	100

206	4.4.14	charset-configured (charset) .....	101
207	4.4.15	charset-supported (1setOf charset) .....	101
208	4.4.16	natural-language-configured (naturalLanguage).....	101
209	4.4.17	generated-natural-language-supported (1setOf naturalLanguage).....	101
210	4.4.18	document-format-default (mimeMediaType).....	102
211	4.4.19	document-format-supported (1setOf mimeMediaType).....	102
212	4.4.20	printer-is-accepting-jobs (boolean) .....	102
213	4.4.21	queued-job-count (integer(0:MAX)).....	102
214	4.4.22	printer-message-from-operator (text(127)) .....	103
215	4.4.23	color-supported (boolean) .....	103
216	4.4.24	reference-uri-schemes-supported (1setOf uriScheme) .....	103
217	4.4.25	pdl-override-supported (type2 keyword) .....	103
218	4.4.26	printer-up-time (integer(1:MAX)) .....	104
219	4.4.27	printer-current-time (dateTime).....	104
220	4.4.28	multiple-operation-time-out (integer(1:MAX)) .....	104
221	4.4.29	compression-supported (1setOf type3 keyword).....	104
222	4.4.30	job-k-octets-supported (rangeOfInteger(0:MAX)).....	105
223	4.4.31	job-impressions-supported (rangeOfInteger(0:MAX)) .....	105
224	4.4.32	job-media-sheets-supported (rangeOfInteger(0:MAX)) .....	105
225	5.	Conformance .....	105
226	5.1	Client Conformance Requirements .....	105
227	5.2	IPP Object Conformance Requirements.....	106
228	5.2.1	Objects.....	106
229	5.2.2	Operations .....	106
230	5.2.3	IPP Object Attributes .....	107
231	5.2.4	Extensions.....	107
232	5.2.5	Attribute Syntaxes.....	108
233	5.3	Charset and Natural Language Requirements .....	108
234	5.4	Security Conformance Requirements.....	108
235	6.	IANA Considerations (registered and private extensions).....	108
236	6.1	Typed 'keyword' and 'enum' Extensions .....	109
237	6.2	Attribute Extensibility .....	111
238	6.3	Attribute Syntax Extensibility .....	112
239	6.4	Operation Extensibility .....	112
240	6.5	Status Code Extensibility .....	112
241	6.6	Registration of MIME types/sub-types for document-formats.....	113
242	6.7	Registration of charsets for use in 'charset' attribute values .....	113
243	7.	Internationalization Considerations .....	113
244	8.	Security Considerations .....	116
245	8.1	Security Scenarios.....	117

246	8.1.1	Client and Server in the Same Security Domain .....	117
247	8.1.2	Client and Server in Different Security Domains .....	117
248	8.1.3	Print by Reference .....	118
249	8.2	URIs for TLS and non-TLS Access.....	118
250	8.3	The "requesting-user-name" (name(MAX)) Operation Attribute.....	118
251	8.4	Restricted Queries.....	120
252	8.5	IPP Security Application Profile for TLS.....	120
253	9.	References.....	121
254	10.	Copyright Notice .....	124
255	11.	Author's Address .....	125
256	12.	Formats for IPP Registration Proposals .....	129
257	12.1	Type2 keyword attribute values registration .....	129
258	12.2	Type3 keyword attribute values registration .....	129
259	12.3	Type2 enum attribute values registration .....	130
260	12.4	Type3 enum attribute values registration .....	130
261	12.5	Attribute registration.....	130
262	12.6	Attribute Syntax registration .....	131
263	12.7	Operation registration .....	131
264	12.8	Status code registration.....	131
265	13.	APPENDIX A: Terminology .....	132
266	13.1	Conformance Terminology.....	132
267	13.1.1	NEED NOT .....	133
268	13.2	Model Terminology .....	133
269	13.2.1	Keyword.....	133
270	13.2.2	Attributes.....	133
271	13.2.2.1	Attribute Name.....	134
272	13.2.2.2	Attribute Group Name.....	134
273	13.2.2.3	Attribute Value.....	134
274	13.2.2.4	Attribute Syntax .....	134
275	13.2.3	Supports .....	134
276	13.2.4	print-stream page .....	136
277	13.2.5	impression.....	136
278	14.	APPENDIX B: Status Codes and Suggested Status Code Messages .....	136
279	14.1	Status Codes.....	137
280	14.1.1	Informational.....	137
281	14.1.2	Successful Status Codes .....	138
282	14.1.2.1	successful-ok (0x0000).....	138
283	14.1.2.2	successful-ok-ignored-or-substituted-attributes (0x0001).....	138
284	14.1.2.3	successful-ok-conflicting-attributes (0x0002).....	138
285	14.1.3	Redirection Status Codes .....	138



286	14.1.4	Client Error Status Codes.....	138
287	14.1.4.1	client-error-bad-request (0x0400) .....	139
288	14.1.4.2	client-error-forbidden (0x0401) .....	139
289	14.1.4.3	client-error-not-authenticated (0x0402) .....	139
290	14.1.4.4	client-error-not-authorized (0x0403).....	139
291	14.1.4.5	client-error-not-possible (0x0404) .....	139
292	14.1.4.6	client-error-timeout (0x0405) .....	139
293	14.1.4.7	client-error-not-found (0x0406).....	140
294	14.1.4.8	client-error-gone (0x0407).....	140
295	14.1.4.9	client-error-request-entity-too-large (0x0408).....	140
296	14.1.4.10	client-error-request-value-too-long (0x0409).....	141
297	14.1.4.11	client-error-document-format-not-supported (0x040A).....	141
298	14.1.4.12	client-error-attributes-or-values-not-supported (0x040B).....	141
299	14.1.4.13	client-error-uri-scheme-not-supported (0x040C).....	142
300	14.1.4.14	client-error-charset-not-supported (0x040D) .....	142
301	14.1.4.15	client-error-conflicting-attributes (0x040E).....	142
302	14.1.5	Server Error Status Codes.....	142
303	14.1.5.1	server-error-internal-error (0x0500).....	142
304	14.1.5.2	server-error-operation-not-supported (0x0501) .....	142
305	14.1.5.3	server-error-service-unavailable (0x0502) .....	142
306	14.1.5.4	server-error-version-not-supported (0x0503).....	143
307	14.1.5.5	server-error-device-error (0x0504) .....	143
308	14.1.5.6	server-error-temporary-error (0x0505) .....	143
309	14.1.5.7	server-error-not-accepting-jobs (0x0506).....	144
310	14.1.5.8	server-error-busy (0x0507).....	144
311	14.2	Status Codes for IPP Operations .....	145
312	15.	APPENDIX C: "media" keyword values .....	145
313	16.	APPENDIX D: Processing IPP Attributes .....	150
314	16.1	Fidelity.....	150
315	16.2	Page Description Language (PDL) Override.....	151
316	16.3	Suggested Operation Processing Steps for All Operations .....	153
317	16.3.1	Validate version number.....	153
318	16.3.2	Validate operation identifier .....	154
319	16.3.3	Validate the request identifier .....	154
320	16.3.4	Validate attribute group and attribute presence and order .....	154
321	16.3.4.1	Validate the presence and order of attribute groups.....	154
322	16.3.4.2	Ignore unknown attribute groups in the expected position.....	154
323	16.3.4.3	Validate the presence of a single occurrence of required Operation attributes.....	155
324	16.3.5	Validate the values of the REQUIRED Operation attributes .....	161
325	16.3.6	Validate the values of the OPTIONAL Operation attributes.....	164

326	16.4	Suggested Additional Processing Steps for Operations that Create/Validate Jobs and	
327		Add Documents .....	166
328	16.4.1	Default "ipp-attribute-fidelity" if not supplied .....	166
329	16.4.2	Check that the Printer object is accepting jobs .....	166
330	16.4.3	Validate the values of the Job Template attributes .....	166
331	16.4.4	Check for conflicting Job Template attributes values.....	170
332	16.4.5	Decide whether to REJECT the request.....	171
333	16.4.6	For the Validate-Job operation, RETURN one of the success status codes .....	171
334	16.4.7	Create the Job object with attributes to support .....	171
335	16.4.8	Return one of the success status codes .....	173
336	16.4.9	Accept appended Document Content.....	173
337	16.4.10	Scheduling and Starting to Process the Job.....	173
338	16.4.11	Completing the Job .....	174
339	16.4.12	Destroying the Job after completion .....	174
340	16.4.13	Interaction with "ipp-attribute-fidelity" .....	174
341	16.5	Using Job Template Attributes During Document Processing.....	174
342	17.	APPENDIX E: Generic Directory Schema.....	176

## 343 1. Introduction

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

348 [Requirements-Design Goals](#) for an Internet Printing Protocol [IPP-REQ] ([informational](#))  
 349 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [IPP-RAT]  
 350 ([informational](#))  
 351 Internet Printing Protocol/1.0: Model and Semantics (this document)  
 352 Internet Printing Protocol/1.0: [Encoding and Transport-and-Encoding](#) [IPP-PRO]  
 353 [Mapping between LPD and IPP Protocols](#) [IPP-LPD] ([informational](#))  
 354  
 355

356 Anyone reading this document for the first time is strongly encouraged to read the IPP documents in the  
 357 following order:

- 358 1. The [requirements design goals](#) document, "[Requirements-Design Goals](#) for an Internet Printing  
 359 Protocol". That document takes a broad look at distributed printing functionality, and it  
 360 enumerates real-life scenarios that help to clarify the features that need to be included in a printing  
 361 protocol for the Internet. It identifies requirements for three types of users: end users, operators,

- 362 and administrators. The requirements-design goals document calls out a subset of end user  
363 requirements that are satisfied in IPP/1.0. Operator and administrator requirements are out of  
364 scope for version 1.0.
- 365 2. The rationale document, "Rationale for the Structure and Model and Protocol for the Internet  
366 Printing Protocol". That document describes IPP from a high level view, defines a roadmap for  
367 the various documents that form the suite of IPP specifications, and gives background and  
368 rationale for the IETF working group's major decisions.
- 369 3. This document, the "Internet Printing Protocol/1.0: Model and Semantics" document. This  
370 document describes a simplified model with abstract objects, their attributes, and their operations.  
371 The model introduces a Printer and a Job. A Job optionally supports multiple documents per Job.  
372 The model document also describes how security, internationalization, and directory issues are  
373 addressed.
- 374 4. The protocol specification, " Internet Printing Protocol/1.0: ~~Transport and~~Encoding and  
375 Transport". That document defines the encoding rules for a new Internet media type called  
376 "application/ipp" and shows a is-a formal mapping of the abstract operations and attributes  
377 defined in the model document onto HTTP/1.1.

378  
379 The LPD mapping document, "Mapping between LPD and IPP Protocols", is an informational document  
380 that recommends a mapping between the commands and operands of IPP. The LPD mapping document  
381 gives some advice to implementers of gateways between IPP and LPD (Line Printer Daemon)  
382 implementations.~~The protocol specification defines the encoding rules for a new Internet media type~~  
383 ~~called "application/ipp".~~

384 This document is laid out as follows:

- 385 - The rest of Section 1 is an introduction to the IPP simplified model for distributed printing.  
386 - Section 2 introduces the object types covered in the model with their basic behaviors, attributes, and  
387 interactions.  
388 - Section 3 defines the operations included in IPP/1.0. IPP operations are synchronous, therefore, for  
389 each operation, there is a both request and a response.  
390 - Section 4 defines the attributes (and their syntaxes) that are used in the model.  
391 - Sections 5 - 6 summarizes the implementation conformance requirements for objects that support  
392 the protocol and IANA considerations, respectively.  
393 - Sections 7 - 11 cover the Internationalization and Security considerations as well as References,  
394 Copyright Notice, and Author contact information.  
395 - Sections 12 - 14 are appendices that cover Terminology, Status Codes and Messages, and "media"  
396 keyword values. This document uses terms such as "attributes", "keywords", and "support".  
397 These terms have special meaning and are defined in the model terminology section. Capitalized  
398 terms, such as MANDATORY, SHALLMUST, and OPTIONAL, have special meaning relating

- 399 to conformance. These terms are defined in the section on conformance terminology, most of  
400 which is taken from RFC 2119 [RFC2119].
- 401 - Section 15 is an appendix that defines the rules and suggested techniques for the processing of  
402 attributes in client requests by IPP objects. This section helps to clarify the effects of interactions  
403 between related attributes and their values.
  - 404 - Section 16 is an appendix that enumerates the subset of Printer attributes that form a generic  
405 directory schema. These attributes are useful when registering a Printer so that a client can find  
406 the Printer not just by name, but by filtered searches as well.

## 407 1.1 Simplified Printing Model

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

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

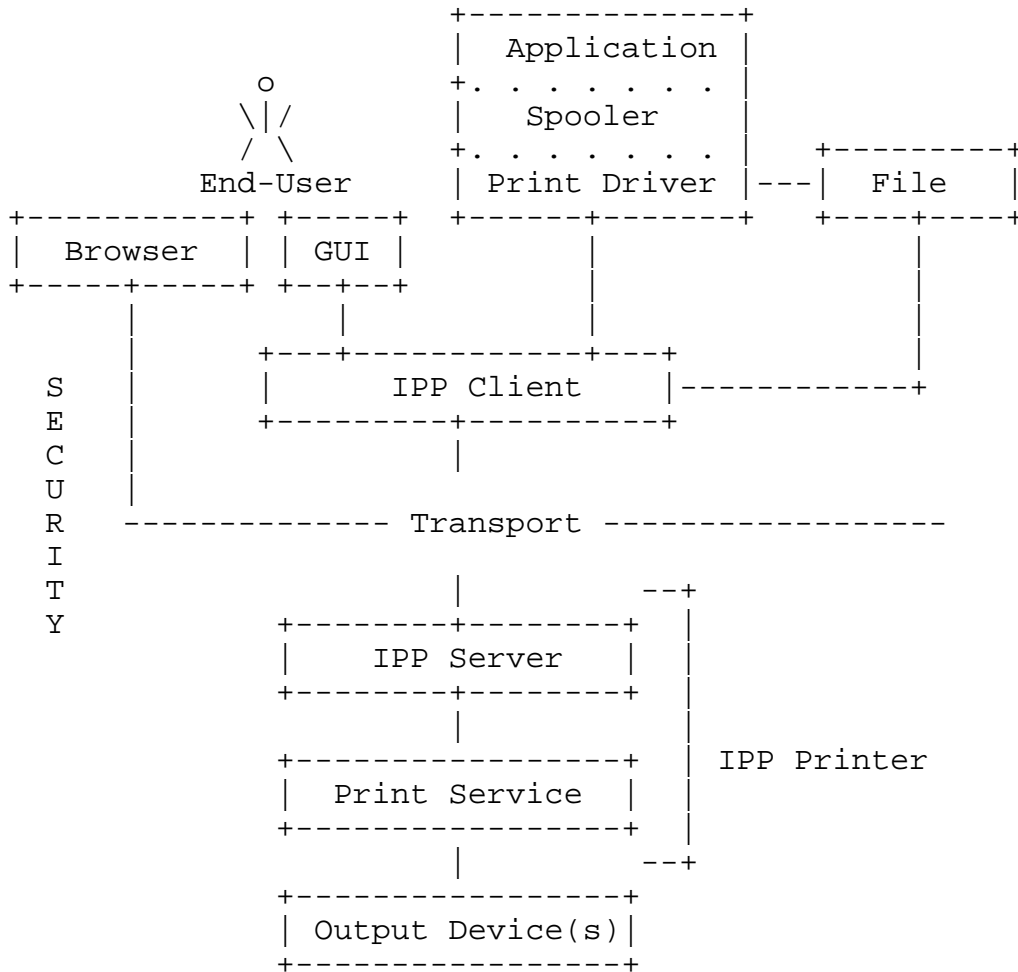
- 420 - Printer (Section 2.1)
- 421 - Job (Section 2.2)

422  
423 Each object type has an associated set of operations (see section 3) and attributes (see section 4).

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

428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458

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

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

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

## 481 2. IPP Objects

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

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

- 491 - "~~MANDATORYREQUIRED~~": each object ~~SHALLMUST~~ support the attribute.
- 492 - "OPTIONAL": each object MAY support the attribute.

493

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

### 496 2.1 Printer Object

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

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

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

509  
510 Since a Printer object is an abstraction of a generic document output device and print service provider, a  
511 Printer object could be used to represent any real or virtual device with semantics consistent with the  
512 Printer object, such as a fax device, an imager, or even a CD writer.

513 Some examples of configurations supporting a Printer object include:

- 514 1) An output device with no spooling capabilities
- 515 2) An output device with a built-in spooler
- 516 3) A print server supporting IPP with one or more associated output devices
  - 517 3a) The associated output devices may or may not be capable of spooling jobs
  - 518 3b) The associated output devices may or may not support IPP

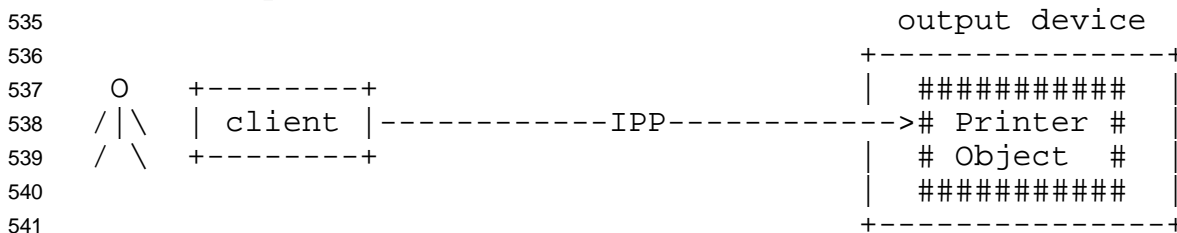
519  
520 The following figures show some examples of how Printer objects can be realized on top of various  
521 distributed printing configurations. The embedded case below represents configurations 1 and 2. The  
522 hosted and fan-out figures below represent configurations 3a and 3b.

523 Legend:

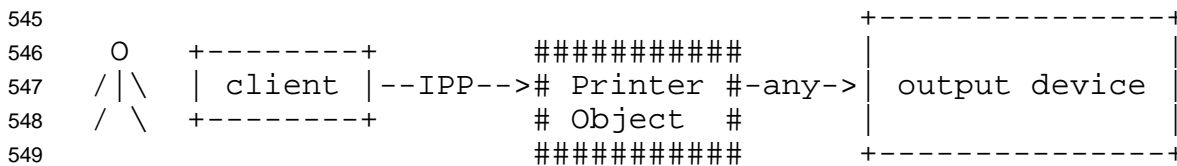
524  
 525 ##### indicates a Printer object which is  
 526 either embedded in an output device or is  
 527 hosted in a server. The Printer object  
 528 might or might not be capable of queuing/spooling.  
 529

530 any indicates any network protocol or direct  
 531 connect, including IPP

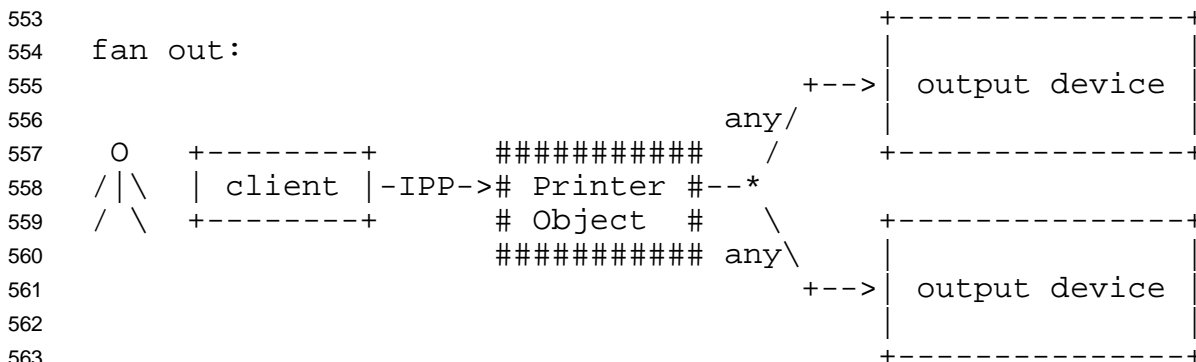
532  
 533  
 534 embedded printer:



542  
 543  
 544 hosted printer:



551  
 552  
 553  
 554 fan out:



564  
 565  
 566 2.2 Job Object



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

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

- 573 - "job-template" attributes: These attributes ~~are~~ can be supplied by the client or end user and include  
574 job processing instructions which are intended to override any Printer object defaults and/or  
575 instructions embedded within the document data. (See section 4.2)
- 576 - "job-description" attributes: These attributes describe the Job object's identification, state, size, etc.  
577 The client supplies some of these attributes, and the Printer object generates others. (See section  
578 4.3)

579  
580 A Job object contains at least one document, but may contain multiple documents. A document is either:

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

584  
585 In IPP/1.0, a document is not modeled as an IPP object, therefore it has no object identifier or associated  
586 attributes. All job processing instructions are modeled as Job object attributes. These attributes are  
587 called Job Template attributes and they apply equally to all documents within a Job object.

## 588 2.3 Object Relationships

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

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

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

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

#### 604 2.4 Object Identity

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

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

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

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

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

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

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

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

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

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

674 In addition to identifiers, Printer objects and Job objects have names ("printer-name" and "job-name").  
675 An object name ~~need not~~**NEED NOT** be unique across all instances of all objects. A Printer object's name  
676 is chosen and set by an administrator through some mechanism outside the scope of IPP/1.0. A Job  
677 object's name is optionally chosen and supplied by the IPP client submitting the job. If the client does not  
678 supply a Job object name, the Printer object generates a name for the new Job object. In all cases, the  
679 name only has local meaning.

680 To summarize:

- 681 - Each Printer object is identified with one or more URIs. The Printer's "printer-uri-supported"  
682 attribute contains the URI(s).
- 683 - The Printer object's "uri-security-supported" attribute identifies the communication channel security  
684 protocols that may or may not have been configured for the various Printer object URIs (e.g., 'tls'  
685 or 'none').
- 686 - Each Job object is identified with a Job URI. The Job's "job-uri" attribute contains the URI.
- 687 - Each Job object is also identified with Job ID which is a 32-bit, positive integer. The Job's "job-id"  
688 attribute contains the Job ID. The Job ID is only unique within the context of the Printer object  
689 which created the Job object.
- 690 - Each Job object has a "job-printer-uri" attribute which contains the URI of the Printer object that  
691 was used to create the Job object. This attribute is used to determine the Printer object that  
692 created a Job object when given only the URI for the Job object. This linkage is necessary to  
693 determine the languages, charsets, and operations which are supported on that Job (the basis for  
694 such support comes from the creating Printer object).
- 695 - Each Printer object has a name (which is not necessarily unique). The administrator chooses and  
696 sets this name through some mechanism outside the scope of IPP/1.0 itself. The Printer object's  
697 "printer-name" attribute contains the name.
- 698 - Each Job object has a name (which is not necessarily unique). The client optionally supplies this  
699 name in the create request. If the client does not supply this name, the Printer object generates a  
700 name for the Job object. The Job object's "job-name" attribute contains the name.

### 701 3. IPP Operations

702 IPP objects support operations. An operation consists of a request and a response. When a client  
703 communicates with an IPP object, the client issues an operation request to the URI for that object.  
704 Operation requests and responses have parameters that identify the operation. Operations also have

705 attributes that affect the run-time characteristics of the operation (the intended target, localization  
706 information, etc.) supply information about the operation itself. These operation-specific attributes are  
707 called operation attributes (as compared to object attributes such as Printer object attributes or Job object  
708 attributes). Each request carries along with it any operation attributes, object attributes, and/or document  
709 data required to perform the operation. Each request requires a response from the object. Each response  
710 indicates success or failure of the operation with a status code as a response parameter. The response  
711 contains any operation attributes, object attributes, and/or status messages generated during the execution  
712 of the operation request.

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

715 The IPP/1.0 Printer operations are:

- 716 Print-Job (section 3.2.1)
- 717 Print-URI (section 3.2.2)
- 718 Validate-Job (section 3.2.3)
- 719 Create-Job (section 3.2.4)
- 720 Get-Printer-Attributes (section 3.2.5)
- 721 Get-Jobs (section 3.2.6)

722

723 The Job operations are:

- 724 Send-Document (section 3.3.1)
- 725 Send-URI (section 3.3.2)
- 726 Cancel-Job (section 3.3.3)
- 727 Get-Job-Attributes (section 3.3.4)

728

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

### 731 3.1 Common Semantics

732 All IPP operations require some share some common parameters and operation attributes, elements and  
733 features. . These common elements and their semantic characteristics are defined and described in more  
734 detail in the following sections.

#### 735 3.1.1 ~~Required Elements~~ Required Parameters

736 Every operation request contains the following REQUIRED parameters:

- 737 - a "version-number",
- 738 - an "operation-id",
- 739 - a "request-id", and
- 740 - the attributes that are MANDATORYREQUIRED for that type of request.

741

742 Every operation response contains the following REQUIRED parameters:

- 743 - a "version-number",
- 744 - a "status-code",
- 745 - the "request-id" that was supplied in the corresponding request, and
- 746 - the attributes that are MANDATORYREQUIRED for that type of response.

747

748 ~~Note:—The transport and encoding and transport~~ document [IPP-PRO] defines special rules for the  
749 encoding of ~~the these parameters. "operation-id", the "version number", the "status code", and the~~  
750 ~~"request-id"~~. All other operation elements are represented using the more generic encoding rules for  
751 attributes and groups of attributes.

### 752 3.1.2 Operation IDs and Request IDs

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

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

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

### 766 3.1.3 Attributes

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



- 769 - Operation Attributes: These attributes are passed in the operation and affect the IPP object's  
770 behavior while processing the operation request and may affect other attributes or groups of  
771 attributes. Some operation attributes describe the document data associated with the print job  
772 and are associated with new Job objects, however most operation attributes do not persist beyond  
773 the life of the operation. The description of each operation attribute includes conformance  
774 statements indicating which operation attributes are **MANDATORYREQUIRED** and which are  
775 **OPTIONAL** for an IPP object to support and which attributes a client **MUST** supply in a request  
776 and an IPP object **MUST** supply in a response.
- 777 - Job Template Attributes: These attributes affect the processing of a job. A client **OPTIONALLY**  
778 supplies Job Template Attributes in a create request, and the receiving object **MUST** be prepared  
779 to receive all supported attributes. The Job object can later be queried to find out what Job  
780 Template attributes were originally requested in the create request, and such attributes are  
781 returned in the response as Job Object Attributes. The Printer object can be queried about its Job  
782 Template attributes to find out what type of job processing capabilities are supported and/or what  
783 the default job processing behaviors are, though such attributes are returned in the response as  
784 Printer Object Attributes. The "ipp-attribute-fidelity" operation attribute affects processing of all  
785 client supplied Job Template attributes (see section 16 for a full description of "ipp-attribute-  
786 fidelity" and its relationship to other attributes).
- 787 - Job Object Attributes: These attributes are returned in response to a query operation directed at a  
788 Job object.
- 789 - Printer Object Attributes: These attributes are returned in response to a query operation directed at  
790 a Printer object.
- 791 - Unsupported Attributes: In a create request, the client supplies a set of Operation and Job Template  
792 attributes. If any of these attributes or their values are unsupported by the Printer object, the  
793 Printer object returns the set of unsupported attributes in the response. Section 16 gives a full  
794 description of how Job Template attributes supplied by the client in a create request are processed  
795 by the Printer object and how unsupported attributes are returned to the client. Because of  
796 extensibility, any IPP object might receive a request that contains new or unknown attributes or  
797 values for which it has no support. In such cases, the IPP object processes what it can and returns  
798 the unsupported attributes in the response.

799  
800 Later in this section, each operation is formally defined by identifying the allowed and expected groups of  
801 attributes for each request and response. The model identifies a specific order for each group in each  
802 request or response, but the attributes within each group may be in any order, unless specified otherwise.

803 Each attribute specification includes the attribute's name followed by the name of its attribute syntax(es)  
804 in parentheses. In addition, each 'integer' attribute is followed by the allowed range in parentheses,  
805 (m:n), for values of that attribute. Each 'text' or 'name' attribute is followed by the maximum size in  
806 octets in parentheses, (size), for values of that attribute. For more details on attribute syntax notation, see  
807 the descriptions of these attributes syntaxes in section 4.1. ~~It is an operational error for clients to supply~~

808 ~~in operation requests and/or IPP objects to returns in operations responses attribute value(s) that do not~~  
809 ~~match the syntax(es) defined for that attribute (see section 3 for operation attributes and section 4 for IPP~~  
810 ~~object attributes).~~

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

815 Note: Some operations are **MANDATORYREQUIRED** for IPP objects to support; the others are  
816 OPTIONAL (see section 5.2.2). Therefore, before using an OPTIONAL operation, a client SHOULD  
817 first use the **MANDATORYREQUIRED** Get-Printer-Attributes operation to query the Printer's  
818 "operations-supported" attribute in order to determine which OPTIONAL Printer and Job operations are  
819 actually supported. The client SHOULD NOT use an OPTIONAL operation that is not supported.  
820 When an IPP object receives a request to perform an operation it does not support, it returns the 'server-  
821 error-operation-not-supported' status code (see section 14.1.5.2). An IPP object is non-conformant if it  
822 does not support a **MANDATORYREQUIRED** operation.

#### 823 3.1.4 Character Set and Natural Language Operation Attributes

824 Some Job and Printer attributes have values that are text strings and names intended for human  
825 understanding rather than machine understanding (see the 'text' and 'name' attribute syntax descriptions in  
826 section 4.1). The following sections describe two special Operation Attributes called "attributes-charset"  
827 and "attributes-natural-language". These attributes are always part of the Operation Attributes group.  
828 For most attribute groups, the order of the attributes within the group is not important. However, for  
829 these two attributes within the Operation Attributes group, the order is critical. The "attributes-charset"  
830 attribute MUST be the first attribute in the group and the "attributes-natural-language" attribute MUST  
831 be the second attribute in the group. In other words, these attributes MUST be supplied in every IPP  
832 request and response, they MUST come first in the group, and MUST come in the specified order. For  
833 job creation operations, the IPP Printer implementation saves these two attributes with the new Job  
834 object as Job Description attributes. For the sake of brevity in this document, these operation attribute  
835 descriptions are not repeated with every operation request and response, but have a reference back to this  
836 section instead.

##### 837 3.1.4.1 Request Operation Attributes

838 The client **SHALLMUST** supply and the Printer object **SHALLMUST** support the following  
839 **MANDATORYREQUIRED** operation attributes in every IPP/1.0 operation request:



840 "attributes-charset" (charset):

841 This operation attribute identifies the charset (coded character set and encoding method) used by  
842 any 'text' and 'name' attributes that the client is supplying in this request. It also identifies the  
843 charset that the Printer object **SHALLMUST** use (if supported) for all 'text' and 'name' attributes  
844 and status messages that the Printer object returns in the response to this request. See Sections  
845 4.1.1 and 4.1.2 for the specification of the 'text' and 'name' attribute syntaxes.

846  
847 All IPP objects **SHALLMUST** support the 'utf-8' charset [RFC2044] and MAY support  
848 additional charsets provided that they are registered with IANA [IANA-CS]. If the Printer object  
849 does not support the client supplied charset value, the Printer object **SHALLMUST** reject the  
850 request and return the 'client-error-charset-not-supported' status code. The Printer object  
851 **SHALLMUST** indicate the charset(s) supported as the values of the "charset-supported" Printer  
852 attribute (see Section 4.4.15), so that the client can query to determine which charset(s) are  
853 supported.

854  
855 Note to client implementers: Since IPP objects are only required to support the 'utf-8' charset, in  
856 order to maximize interoperability with multiple IPP object implementations, a client may want to  
857 supply 'utf-8' in the "attributes-charset" operation attribute, even though the client is only passing  
858 and able to present a simpler charset, such as US-ASCII or ISO-8859-1. Then the client will have  
859 to filter out (or charset convert) those characters that are returned in the response that it cannot  
860 present to its user. On the other hand, if both the client and the IPP objects also support a charset  
861 in common besides utf-8, the client may want to use that charset in order to avoid charset  
862 conversion or data loss.

863  
864 See the 'charset' attribute syntax description in Section 4.1.7 for the syntax and semantic  
865 interpretation of the values of this attribute and for example values.

866  
867 "attributes-natural-language" (naturalLanguage):

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

872  
873 There are no **MANDATORYREQUIRED** natural languages required for the Printer object to  
874 support. However, the Printer object's "generated-natural-language-supported" attribute  
875 identifies the natural languages supported by the Printer object and any contained Job objects for  
876 all text strings generated by the IPP object. A client MAY query this attribute to determine which  
877 natural language(s) are supported for generated messages.

878

879 For any of the attributes for which the Printer object generates text, i.e., for the "job-state-  
880 message", "printer-state-message", and status messages (see Section 3.1.6), the Printer object  
881 **SHALLMUST** be able to generate these text strings in any of its supported natural languages. If  
882 the client requests a natural language that is not supported, the Printer object **SHALLMUST**  
883 return these generated messages in the Printer's configured natural language as specified by the  
884 Printer's "natural-language-configured" attribute" (see Section 4.4.16).

885  
886 For other 'text' and 'name' attributes supplied by the client, authentication system, operator,  
887 system administrator, or manufacturer, i.e., for "job-originating-user-name", "printer-name"  
888 (name), "printer-location" (text), "printer-info" (text), and "printer-make-and-model" (text), the  
889 Printer object is only required to support the configured natural language of the Printer identified  
890 by the Printer object's "natural-language-configured" attribute, though support of additional  
891 natural languages for these attributes is permitted.

892  
893 For any 'text' or 'name' attribute in the request that is in a different natural language than the value  
894 supplied in the "attributes-natural-language", the client **SHALLMUST** use the Natural Language  
895 Override mechanism (see sections 4.1.1.2 and 4.1.2.2) for each such attribute value supplied.

896  
897 The IPP object **SHALLMUST** accept any natural language and any Natural Language Override,  
898 whether the IPP object supports that natural language or not (and independent of the value of the  
899 "ipp-attribute-fidelity" Operation attribute). That is the IPP object accepts all client supplied  
900 values no matter what the values are in the Printer object's "generated-natural-language-  
901 supported" attribute. That attribute, "generated-natural-language-supported", only applies to  
902 generated messages, not client supplied messages. The IPP object **SHALLMUST** remember that  
903 natural language for all client supplied attributes, and when returning those attributes in response  
904 to a query, the IPP object **SHALLMUST** indicate that natural language.

905  
906 For example, the "job-name" attribute MAY be supplied by the client in a create request. The text  
907 value for this attribute will be in the natural language identified by the "attribute-natural-language"  
908 attribute, or if different, as identified by the Natural Language Override mechanism. If supplied,  
909 the IPP object will use the value of the "job-name" attribute to populate the Job object's "job-  
910 name" attribute. Whenever any client queries the Job object's "job-name" attribute, the IPP object  
911 returns the attribute as stored and uses the Natural Language Override mechanism to specify the  
912 natural language, if it is different from that reported in the "attributes-natural-language" operation  
913 attribute of the response. An IPP object **SHALLMUST** NOT reject a request based on a supplied  
914 natural language in an "attributes-natural-language" Operation attribute or in any attribute that  
915 uses the Natural Language Override.

916  
917 See the 'naturalLanguage' attribute syntax description in section 4.1.8 for the syntax and semantic  
918 interpretation of the values of this attribute and for example values.

919

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

926 - In a create request: -If a the client supplies a text or name attribute (for example, the "job-name"  
927 operation attribute) that uses that specific invalid-an apparently incompatible combination, it is a  
928 client choice that -and it does not n't affect the Printer object or its correct operation -to accept the  
929 invalid combination. Therefore, In this case, the Printer object simply accepts the client supplied  
930 value, stores it with the Job object, and responds back with the same invalid combination  
931 whenever the client (or any client) queries for that attribute.

932 --In a query-type operation, (like Get-Printer-Attributes: for example), -iIf the client requests an  
933 invalid apparently incompatible combination, the Printer object simply responds (as described in  
934 section 3.1.4.2 below) using the Printer's configured natural language rather than the natural  
935 language requested by the client.

936

937 -In either case, the Printer object does not reject the request because of the apparent incompatibility.  
938 The potential an invalid incompatible combination of charset and natural language can occur (either at the  
939 global operation level or at the Natural Language Override attribute-by-attribute level). In addition, since  
940 the response always includes explicit charset and natural language information, there is never any  
941 question or ambiguity in how the client interprets the response.

#### 942 3.1.4.2 Response Operation Attributes

943 The Printer object **SHALLMUST** supply and the client **SHALLMUST** support the following  
944 **MANDATORYREQUIRED** operation attributes in every IPP/1.0 operation response:

945 "attributes-charset" (charset):

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

951

952 If the Printer object supports more than just the 'utf-8' charset, the Printer object **SHALLMUST**  
953 be able to code convert between each of the charsets supported on a highest fidelity possible basis  
954 in order to return the 'text' and 'name' attributes in the charset requested by the client. However,

955 some information loss MAY occur during the charset conversion depending on the charsets  
956 involved. For example, the Printer object may convert from a UTF-8 'a' to a US-ASCII 'a' (with  
957 no loss of information), from an ISO Latin 1 CAPITAL LETTER A WITH ACUTE ACCENT  
958 to US-ASCII 'A' (losing the accent), or from a UTF-8 Japanese Kanji character to some ISO  
959 Latin 1 error character indication such as '?', decimal code equivalent, or to the absence of a  
960 character, depending on implementation.

961  
962 Note: Whether an implementation that supports more than one charset stores the data in the  
963 charset supplied by the client or code converts to one of the other supported charsets, depends on  
964 implementation. The strategy should try to minimize loss of information during code conversion.  
965 On each response, such an implementation converts from its internal charset to that requested.  
966

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

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

### 977 3.1.5 Operation Targets

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

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

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

991

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

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

- 999 - In the case where there is only one operation target attribute (i.e., either only the "printer-uri"  
1000 attribute or only the "job-uri" attribute), that attribute MUST be the third attribute in the  
1001 operation attributes group.
- 1002 - In the case where Job operations use two operation target attributes (i.e., the "printer-uri" and "job-  
1003 id" attributes), the "printer-uri" attribute MUST be the third attribute and the "job-id" attribute  
1004 MUST be the fourth attribute.

1005  
1006 In all cases, the target URIs contained within the body of IPP operation requests and responses must be  
1007 in absolute format rather than relative format (a relative URL identifies a resource with the scope of the  
1008 HTTP server, but does not include scheme, host or port).

1009 ~~Note: The IPP transport and encoding document [IPP-PRO] calls for the target URL to be included both~~  
1010 ~~inside the IPP operation (as MANDATORY operation attributes) and outside the operation (at the HTTP~~  
1011 ~~layer). The potential exists that these two values reference the same IPP object, but are not literally~~  
1012 ~~identical since one can be a relative URL and the other can be an absolute URL. HTTP/1.1 allows clients~~  
1013 ~~to generate and send a relative URL rather than an absolute URL. A relative URL identifies a resource~~  
1014 ~~with the scope of the HTTP server, but does not include scheme, host or port. The following statements~~  
1015 ~~characterize how URLs should be used in the mapping of IPP onto HTTP/1.1:~~

- 1016 ~~1. Although potentially redundant, a client MUST supply the target of the operation both as an~~  
1017 ~~Operation Attribute (see Section 3.1.5) and as a URL at the HTTP layer. The rationale for this~~  
1018 ~~decision is to maintain a consistent set of rules for mapping IPP to possibly many communication~~  
1019 ~~layers, even where URLs are not used as the addressing mechanism.~~
- 1020 ~~2. Even though these two URLs might not be literally identical (one being relative and the other being~~  
1021 ~~absolute), they must both reference the same IPP object.~~
- 1022 ~~3. The URL in the HTTP layer is either relative or absolute and is used by the HTTP server to route~~  
1023 ~~the HTTP request to the correct resource relative to that HTTP server. The HTTP server need~~  
1024 ~~not be aware of the URL within the operation request.~~
- 1025 ~~4. Once the HTTP server resource begins to process the HTTP request, it might get the reference to~~  
1026 ~~the appropriate IPP Printer object from either the HTTP URL (using to the context of the HTTP~~  
1027 ~~server for relative URLs) or from the URL within the operation request; the choice is up to the~~  
1028 ~~implementation.~~

1029 ~~5. HTTP URLs can be relative or absolute, but the target URL in the operation MUST be an absolute~~  
1030 ~~URL~~

1031  
1032 The following rules apply to the use of port numbers in URIs that identify IPP objects:

- 1033 1. If the URI scheme allows the port number to be explicitly included in the URI string, and a port  
1034 number is specified within the URI, then that port number MUST be used by the client to contact  
1035 the IPP object.
- 1036  
1037 2. If the URI scheme allows the port number to be explicitly included in the URI string, and a port  
1038 number is not specified within the URI, then default port number implied by that URI scheme  
1039 MUST be used by the client to contact the IPP object.
- 1040  
1041 3. If the ~~—~~URI scheme does not allow an explicit port number to be specified within the URI, then  
1042 the default port number implied by that URI MUST be used by the client to contact the IPP  
1043 object.

1044  
1045 Note: The IPP encoding and transport ~~and encoding~~ document [IPP-PRO] shows a mapping of IPP onto  
1046 HTTP/1.1 -and defines a new default port number for using IPP over HTTP/1.1.

### 1047 3.1.6 Operation Status Codes and Messages

1048 Every operation response includes a ~~MANDATORYREQUIRED~~ "status-code" parameter and an  
1049 OPTIONAL "status-message" operation attribute. The "status-code" provides information on the  
1050 processing of a request. A "status-message" attribute provides a short textual description of the status of  
1051 the operation. The status code is intended for use by automata, and the status message is intended for the  
1052 human end user. If a response does include a "status-message" attribute, an IPP client NEED NOT  
1053 examine or display the message, however it SHOULD do so in some implementation specific manner.

1054 The "status-code" value is a numeric value that has semantic meaning. The "status-code" syntax is  
1055 similar to a "type2 enum" (see section 4.1 on "Attribute Syntaxes") except that values can range only  
1056 from 0x0000 to 0x7FFF. Section 14 describes the status codes, assigns the numeric values, and suggests  
1057 a corresponding status message for each status code. The "status-message" attribute's syntax is  
1058 "text(255)".

1059 A client implementation of IPP SHOULD convert status code values into any localized message that has  
1060 semantic meaning to the end user. If the Printer object supports the status message, the Printer object  
1061 MUST be able to generate this message in any of the natural languages identified by the Printer object's  
1062 "generated-natural-language-supported" attribute (see the "attributes-natural-language" operation  
1063 attribute specified in section 3.1.4.1). As described in section 3.1.4.1 for any returned 'text' attribute, if  
1064 there is a choice for generating this message, the Printer object uses the natural language indicated by the



1065 value of the "attributes-natural-language" in the client request if supported, otherwise the Printer object  
1066 uses the value in the Printer object's own "natural-language-configured" attribute.

### 1067 3.1.7 Versions

1068 Each operation request and response carries with it a "version-number" parameter. Each value of the  
1069 "version-number" is in the form "X.Y" where X is the major version number and Y is the minor version  
1070 number. By including a version number in the client request, it allows the client to identify which version  
1071 of IPP it is interested in using. If the IPP object does not support that version, the object responds with a  
1072 status code of 'server-error-version-not-supported' along with the closest version number that is  
1073 supported (see section 14.1.5.4).

1074 There is no version negotiation per se. However, if after receiving a 'server-error-version-not-supported'  
1075 status code from an IPP object, there is nothing that prevents a client from trying again with a different  
1076 version number. In order to conform to IPP/1.0, an implementation MUST support at least version '1.0'.

1077 There is only one notion of "version number" that covers both IPP Model and IPP Protocol changes.  
1078 Thus the version number MUST change when introducing a new version of the Model document or a  
1079 new version of the Protocol document.

1080 Changes to the major version number indicate structural or syntactic changes that make it impossible for  
1081 older version of IPP clients and Printer objects to correctly parse and process the new or changed  
1082 attributes, operations and responses. If the major version number changes, the minor version numbers is  
1083 set to zero. As an example, adding the "ipp-attribute-fidelity" attribute (if it had not been part of version  
1084 '1.0'), would have required a change to the major version number. Items that might affect the changing of  
1085 the major version number include any changes to the protocol specification itself, such as:

- 1086 - reordering of ordered attributes or attribute sets
- 1087 - changes to the syntax of existing attributes
- 1088 - changing Operation or Job Template attributes from OPTIONAL to MANDATORYREQUIRED  
1089 and vice versa
- 1090 - adding MANDATORYREQUIRED (for an IPP object to support) operation attributes
- 1091 - adding MANDATORYREQUIRED (for an IPP object to support) operation attribute groups
- 1092 - adding values to existing operation attributes
- 1093 - adding MANDATORYREQUIRED operations

1094  
1095 Changes to the minor version number indicate the addition of new features, attributes and attribute values  
1096 that may not be understood by all IPP objects, but which can be ignored if not understood. Items that  
1097 might affect the changing of the minor version number include any changes to the model objects and  
1098 attributes but not the encoding and transport rules protocol specification-[IPP-PRO]itself (except adding  
1099 attribute syntaxes). Examples of such changes are:-such as:

- 1100 - grouping all extensions not included in a previous version into a new version
- 1101 - adding new attribute values
- 1102 - adding new object attributes
- 1103 - adding OPTIONAL (for an IPP object to support) operation attributes (i.e., those attributes that an
- 1104 IPP object can ignore without confusing clients)
- 1105 - adding OPTIONAL (for an IPP object to support) operation attribute groups (i.e., those attributes
- 1106 that an IPP object can ignore without confusing clients)
- 1107 - adding new attribute syntaxes
- 1108 - adding OPTIONAL operations
- 1109 - changing Job Description attributes or Printer Description attributes from OPTIONAL to
- 1110 ~~MANDATORY~~REQUIRED or vice versa.

1111  
1112 The encoding of the "operation-id", the "version-number", the "status-code", and the "request-id"  
1113 ~~SHALL~~MUST NOT change over any version number (either major or minor). This rule guarantees that  
1114 all future versions will be backwards compatible with all previous versions (at least for checking the  
1115 "operation-id", the "version-number", and the "request-id"). In addition, any protocol elements  
1116 (attributes, error codes, tags, etc.) that are not carried forward from one version to the next are  
1117 deprecated so that they can never be reused with new semantics.

1118 Implementations that support a certain major version NEED NOT support ALL previous versions. As  
1119 each new major version is defined (through the release of a new specification), that major version will  
1120 specify which previous major versions MUST be supported in compliant implementations.

### 1121 3.1.8 Job Creation Operations

1122 In order to "submit a print job" and create a new Job object, a client issues a create request. A create  
1123 request is any one of following three operation requests:

- 1124 - The Print-Job Request: A client that wants to submit a print job with only a single document uses  
1125 the Print-Job operation. The operation allows for the client to "push" the document data to the  
1126 Printer object by including the document data in the request itself.
- 1127  
1128 - The Print-URI Request: A client that wants to submit a print job with only a single document  
1129 (where the Printer object "pulls" the document data instead of the client "pushing" the data to the  
1130 Printer object) uses the Print-URI operation. In this case, the client includes in the request only a  
1131 URI reference to the document data (not the document data itself).
- 1132  
1133 - The Create-Job Request: A client that wants to submit a print job with multiple documents uses the  
1134 Create-Job operation. This operation is followed by an arbitrary number of Send-Document  
1135 and/or Send-URI operations (each creating another document for the newly create Job object).



1136 The Send-Document operation includes the document data in the request (the client "pushes" the  
1137 document data to the printer), and the Send-URI operation includes only a URI reference to the  
1138 document data in the request (the Printer "pulls" the document data from the referenced location).  
1139 The last Send-Document or Send-URI request for a given Job object includes a "last-document"  
1140 operation attribute set to 'true' indicating that this is the last request.

1141

1142 Throughout this model specification, the term "create request" is used to refer to any of these three  
1143 operation requests.

1144 A Create-Job operation followed by only one Send-Document operation is semantically equivalent to a  
1145 Print-Job operation, however, for performance reasons, the client SHOULD use the Print-Job operation  
1146 for all single document jobs. Also, Print-Job is a ~~MANDATORYREQUIRED~~ operation (all  
1147 implementations MUST support it) whereas Create-Job is an OPTIONAL operation, hence some  
1148 implementations might not support it.

1149 Job submission time is the point in time when a client issues a create request. The initial state of every  
1150 Job object is the 'pending' or 'pending-held' state. Later, the Printer object begins processing the print job.  
1151 At this point in time, the Job object's state moves to 'processing'. This is known as job processing time.  
1152 There are validation checks that must be done at job submission time and others that must be performed  
1153 at job processing time.

1154 At job submission time and at the time a Validate-Job operation is received, the Printer MUST do the  
1155 following:

- 1156 1. Process the client supplied attributes and either accept or reject the request
- 1157 2. Validate the syntax of and support for the scheme of any client supplied URI

1158

1159 Section 16 describes the rules and issues surrounding the processing of client supplied attributes. Section  
1160 16.3 presents suggested steps for an IPP object to either accept or reject any request. Section 16.4  
1161 presents suggested additional steps for processing create requests.

1162 At job submission time the Printer ~~SHOULD-NEED~~ NOT perform the validation checks reserved for job  
1163 processing time such as:

- 1164 1. Validating the document data
- 1165 2. Validating the actual contents of any client supplied URI (resolve the reference and follow the link  
1166 to the document data)

1167

1168 At job submission time, these additional job processing time validation checks are essentially useless,  
1169 since they require actually parsing and interpreting the document data, are not guaranteed to be 100%  
1170 accurate, and MUST be done, yet again, at job processing time. Also, in the case of a URI, checking for

1171 availability at job submission time does not guarantee availability at job processing time. In addition, at  
1172 job processing time, the Printer object might discover any of the following conditions that were not  
1173 detectable at job submission time:

- 1174 - runtime errors in the document data,
- 1175 - nested document data that is in an unsupported format,
- 1176 - the URI reference is no longer valid (i.e., the server hosting the document might be down), or
- 1177 - any other job processing error

1178

1179 At job processing time, since the Printer object has already responded with a successful status code in the  
1180 response to the create request, if the Printer object detects an error, the Printer object is unable to inform  
1181 the end user of the error with an operation status code. In this case, the Printer, depending on the error,  
1182 can set the "job-state", "job-state-reasons", or "job-state-message" attributes to the appropriate value(s)  
1183 so that later queries can report the correct job status.

1184 Note: Asynchronous notification of events is outside the scope of IPP/1.0.

## 1185 3.2 Printer Operations

1186 All Printer operations are directed at Printer objects. A client **MUST** always supply the "printer-uri"  
1187 operation attribute in order to identify the correct target of the operation.

### 1188 3.2.1 Print-Job Operation

1189 This **MANDATORYREQUIRED** operation allows a client to submit a print job with only one document  
1190 and supply the document data (rather than just a reference to the data). See Section 16 for the suggested  
1191 steps for processing create operations and their Operation and Job Template attributes.

#### 1192 3.2.1.1 Print-Job Request

1193 The following groups of attributes are supplied as part of the Print-Job Request:

##### 1194 Group 1: Operation Attributes

1195 Natural Language and Character Set:

1196 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1197 3.1.4.1. The Printer object **SHALLMUST** copy these values to the corresponding Job  
1198 Description attributes described in sections 4.3.23 and 4.3.24.

1199

1200 Target:

1201 The "printer-uri" (uri) operation attribute which is the target for this operation as described in  
1202 section 3.1.5.

1203

1204 Requesting User Name:

1205 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1206 described in section 8.3.

1207

1208 "job-name" (name(MAX)):

1209 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1210 attribute. It contains the client supplied Job name. If this attribute is supplied by the client, its  
1211 value is used for the "job-name" attribute of the newly created Job object. The client MAY  
1212 automatically include any information that will help the end-user distinguish amongst his/her jobs,  
1213 such as the name of the application program along with information from the document, such as  
1214 the document name, document subject, or source file name. If this attribute is not supplied by the  
1215 client, the Printer generates a name to use in the "job-name" attribute of the newly created Job  
1216 object (see Section 4.3.5).

1217

1218 "ipp-attribute-fidelity" (boolean):

1219 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1220 attribute. The value 'true' indicates that total fidelity to client supplied Job Template attributes  
1221 and values is required, else the Printer object **SHALLMUST** reject the Print-Job request. The  
1222 value 'false' indicates that a reasonable attempt to print the Job object is acceptable and the Printer  
1223 object **SHALLMUST** accept the Print-job request. If not supplied, the Printer object assumes the  
1224 value is 'false'. All Printer objects MUST support both types of job processing. See section 16  
1225 for a full description of "ipp-attribute-fidelity" and its relationship to other attributes, especially  
1226 the Printer object's "pdl-override-supported" attribute.

1227

1228 "document-name" (name(MAX)):

1229 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1230 attribute. It contains the client supplied document name. The document name MAY be different  
1231 than the Job name. Typically, the client software automatically supplies the document name on  
1232 behalf of the end user by using a file name or an application generated name. If this attribute is  
1233 supplied, its value can be used in a manner defined by each implementation. Examples include:  
1234 printed along with the Job (job start sheet, page adornments, etc.), used by accounting or  
1235 resource tracking management tools, or even stored along with the document as a document level  
1236 attribute. IPP/1.0 does not support the concept of document level attributes.

1237

1238 "document-format" (mimeMediaType) :

1239 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1240 attribute. The value of this attribute identifies the format of the supplied document data. If the  
1241 client does not supply this attribute, the Printer object assumes that the document data is in the  
1242 format defined by the Printer object's "document-format-default" attribute. If the client supplies  
1243 this attribute, but the value is not supported by the Printer object, i.e., the value is not one of the  
1244 values of the Printer object's "document-format-supported" attribute, the Printer object  
1245 **SHALLMUST** reject the request and return the 'client-error-document-format-not-supported'  
1246 status code.

1247

1248 "document-natural-language" (naturalLanguage):

1249 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports  
1250 this attribute. This attribute specifies the natural language of the document for those document-  
1251 formats that require a specification of the natural language in order to image the document  
1252 unambiguously. There are no particular values required for the Printer object to support.

1253

1254 "compression" (type3 keyword)

1255 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports  
1256 this attribute and the "compression-supported" attribute (see section 4.4.29). The client supplied  
1257 "compression" operation attribute identifies the compression algorithm used on the document  
1258 data. If the client omits this attribute, the Printer object **SHALLMUST** assume that the data is  
1259 not compressed. If the client supplies the attribute and the Printer object supports the attribute,  
1260 the Printer object uses the corresponding decompression algorithm on the document data. If the  
1261 client supplies this attribute, but the value is not supported by the Printer object, i.e., the value is  
1262 not one of the values of the Printer object's "compression-supported" attribute, the Printer object  
1263 **SHALLMUST** copy the attribute and its value to the Unsupported Attributes response group,  
1264 reject the request, and return the 'client-error-attributes-or-values-not-supported' status code.

1265

1266 "job-k-octets" (integer(0:MAX))

1267 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports  
1268 this attribute and the "job-k-octets-supported" attribute (see section 4.4.30). The client supplied  
1269 "job-k-octets" operation attribute identifies the total size of the document(s) in K octets being  
1270 submitted (see section 4.3.17 for the complete semantics). If the client supplies the attribute and  
1271 the Printer object supports the attribute, the value of the attribute is used to populate the Job  
1272 object's "job-k-octets" Job Description attribute.

1273

1274 Note: For this attribute and the following two attributes ("job-impressions", and "job-media-  
1275 sheets"), if the client supplies the attribute, but the Printer object does not support the attribute,  
1276 the Printer object ignores the client-supplied value. If the client supplies the attribute and the  
1277 Printer supports the attribute, and the value is within the range of the corresponding Printer

1278 object's "xxx-supported" attribute, the Printer object **SHALLMUST** use the value to populate the  
1279 Job object's "xxx" attribute. If the client supplies the attribute and the Printer supports the  
1280 attribute, but the value is outside the range of the corresponding Printer object's "xxx-supported"  
1281 attribute, the Printer object **SHALLMUST** copy the attribute and its value to the Unsupported  
1282 Attributes response group, reject the request, and return the 'client-error-attributes-or-values-not-  
1283 supported' status code. If the client does not supply the attribute, the Printer object **MAY** choose  
1284 to populate the corresponding Job object attribute depending on whether the Printer object  
1285 supports the attribute and is able to calculate or discern the correct value.

1286

1287 "job-impressions" (integer(0:MAX))

1288 The client **OPTIONALLY** supplies this attribute. The Printer object **OPTIONALLY** supports  
1289 this attribute and the "job-impressions-supported" attribute (see section 4.4.31). The client  
1290 supplied "job-impressions" operation attribute identifies the total size in number of impressions of  
1291 the document(s) being submitted (see section 4.3.18 for the complete semantics).

1292

1293 See note under "job-k-octets".

1294

1295 "job-media-sheets" (integer(0:MAX))

1296 The client **OPTIONALLY** supplies this attribute. The Printer object **OPTIONALLY** supports  
1297 this attribute and the "job-media-sheets-supported" attribute (see section 4.4.32). The client  
1298 supplied "job-media-sheets" operation attribute identifies the total number of media sheets to be  
1299 produced for this job (see section 4.3.19 for the complete semantics).

1300

1301 See note under "job-k-octets".

1302

1303 Group 2: Job Template Attributes

1304 The client **OPTIONALLY** supplies a set of Job Template attributes as defined in section 4.2.

1305

1306 Group 3: Document Content

1307 The client **MUST** supply the document data to be processed.

1308

1309 Note: In addition to the **MANDTORY** ~~common-elements~~parameters required for every operation  
1310 request, the simplest Print-Job Request consists of just the "attributes-charset" and "attributes-natural-  
1311 language" operation attributes; the "printer-uri" target operation attribute; the Document Content and  
1312 nothing else. In this simple case, the Printer object:

1313 - creates a new Job object (the Job object contains a single document),

- 1314 - stores a generated Job name in the "job-name" attribute in the natural language and charset  
1315 requested (see Section 3.1.4.1) (if those are supported, otherwise using the Printer object's default  
1316 natural language and charset), and  
1317 - at job processing time, uses its corresponding default value attributes for the supported Job  
1318 Template attributes that were not supplied by the client as IPP attribute or embedded instructions  
1319 in the document data.  
1320

### 1321 3.2.1.2 Print-Job Response

1322 The Printer object **SHALLMUST** return to the client the following sets of attributes as part of the Print-  
1323 Job Response:

#### 1324 Group 1: Operation Attributes

##### 1325 Status Message:

1326 In addition to the **MANDATORYREQUIRED** status code returned in every response, the  
1327 response **OPTIONALLY** includes a "status-message" (text) operation attribute as described in  
1328 section 3.1.6. If the client supplies unsupported or conflicting Job Template attributes or values,  
1329 the Printer object **SHALLMUST** reject or accept the Print-Job request depending on the whether  
1330 the client supplied a 'true' or 'false' value for the "ipp-attribute-fidelity" operation attribute. See  
1331 section 16 for a complete description of the suggested steps for processing a create request.  
1332

##### 1333 Natural Language and Character Set:

1334 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1335 3.1.4.2.  
1336

#### 1337 Group 2: Unsupported Attributes

1338 This is a set of Operation and Job Template attributes supplied by the client (in the request) that  
1339 are not supported by the Printer object or that conflict with one another (see sections 16.3 and  
1340 16.4).  
1341

1342 Unsupported attributes fall into three categories:  
1343

- 1344 1. The Printer object does not support the named attribute (no matter what the value).
- 1345 2. The Printer object does support the attribute, but does not support some or all of the particular  
1346 values supplied by the client (i.e., the Printer object does not have those values in the  
1347 corresponding supported values attribute).

1348 3. The Printer object does support the attributes and values supplied, but the particular values are  
1349 in conflict with one another, because they violate a constraint, such as not being able to  
1350 staple transparencies.

1351  
1352 In the case of an unsupported attribute name, the Printer object returns the client-supplied  
1353 attribute with a substituted "out-of-band" value of 'unsupported' indicating no support for the  
1354 attribute itself (see the beginning of section 4.1).

1355  
1356 In the case of a supported attribute with one or more unsupported values, the Printer object  
1357 simply returns the client-supplied attribute with the unsupported values as supplied by the client.  
1358 This indicates support for the attribute, but no support for that particular value. If the client  
1359 supplies a multi-valued attribute with more than one value and the Printer object supports the  
1360 attribute but only supports a subset of the client supplied values, the Printer object **SHALLMUST**  
1361 return only those values that are unsupported.

1362  
1363 In the case of two (or more) supported attribute values that are in conflict with one another  
1364 (although each is supported independently, ~~they-the~~ values conflict when requested together  
1365 within the same job), the Printer object **SHALLMUST** return all the values that it ignores or  
1366 substitutes to resolve the conflict, but not any of the values that it is still using. The choice for  
1367 exactly how to resolve the conflict is implementation dependent. See Section 16.4.4 for an  
1368 example.

1369  
1370 In these three cases, the value of the "ipp-attribute-fidelity" supplied by the client does not affect  
1371 what the Printer object returns. The value of "ipp-attribute-fidelity" only affects whether the  
1372 Print-Job operation is accepted or rejected. If the job is accepted, the client may query the job  
1373 using the Get-Job-Attributes operation requesting the unsupported attributes that were returned in  
1374 the create response to see which attributes were ignored (not stored on the Job object) and which  
1375 attributes were stored with other (substituted) values.

### 1376 1377 Group 3: Job Object Attributes

1378 "job-uri" (uri):

1379 The Printer object **MUST** return the Job object's URI by returning the contents of the  
1380 **MANDATORYREQUIRED** "job-uri" Job object attribute. The client uses the Job object's URI  
1381 when directing operations at the Job object. The Printer object always uses its configured security  
1382 policy when creating the new URI. However, if the Printer object supports more than one URI,  
1383 the Printer object also uses information about which URI was used in the Print-Job Request to  
1384 generated the new URI so that the new URI references the correct access channel. In other  
1385 words, if the Print-Job Request comes in over a secure channel, the Printer object **MUST** generate  
1386 a Job URI that uses the secure channel as well.



1387

1388 "job-id" (integer(1:MAX)):

1389 The Printer object MUST return the Job object's Job ID by returning the

1390 **MANDATORYREQUIRED** "job-id" Job object attribute. The client uses this "job-id" attribute

1391 in conjunction with the "printer-uri" attribute used in the Print-Job Request when directing Job

1392 operations at the Printer object.

1393

1394 "job-state":

1395 The Printer object MUST return the Job object's **MANDATORYREQUIRED** "job-state"

1396 attribute. The value of this attribute (along with the value of the next attribute "job-state-reasons")

1397 is taken from a "snapshot" of the new Job object at some meaningful point in time

1398 (implementation defined) between when the Printer object receives the Print-Job Request and

1399 when the Printer object returns the response.

1400

1401 "job-state-reasons":

1402 The Printer object OPTIONALLY returns the Job object's OPTIONAL "job-state-reasons"

1403 attribute. If the Printer object supports this attribute then it MUST be returned in the response.

1404 If this attribute is not returned in the response, the client can assume that the "job-state-reasons"

1405 attribute is not supported and will not be returned in a subsequent Job object query.

1406

1407 "job-state-message":

1408 The Printer object OPTIONALLY returns the Job object's OPTIONAL "job-state-message"

1409 attribute. If the Printer object supports this attribute then it MUST be returned in the response.

1410 If this attribute is not returned in the response, the client can assume that the "job-state-message"

1411 attribute is not supported and will not be returned in a subsequent Job object query.

1412

1413 "number-of-intervening-jobs":

1414 The Printer object OPTIONALLY returns the Job object's OPTIONAL "number-of-intervening-

1415 jobs" attribute. If the Printer object supports this attribute then it MUST be returned in the

1416 response. If this attribute is not returned in the response, the client can assume that the "number-

1417 of-intervening-jobs" attribute is not supported and will not be returned in a subsequent Job object

1418 query.

1419

1420 Note: Since any printer state information which affects a job's state is reflected in the "job-state"

1421 and "job-state-reasons" attributes, it is sufficient to return only these attributes and no specific

1422 printer status attributes.

1423

1424 Note: In addition to the MANDTORY **common-elementsparameters** required for every operation

1425 response, the simplest response consists of the just the "attributes-charset" and "attributes-natural-



1426 language" operation attributes and the "job-uri", "job-id", and "job-state" Job Object Attributes. In this  
1427 simplest case, the status code is "successful-ok" and there is no "status-message" operation attribute.

### 1428 3.2.2 Print-URI Operation

1429 This OPTIONAL operation is identical to the Print-Job operation (section 3.2.1) except that a client  
1430 supplies a URI reference to the document data using the "document-uri" (uri) operation attribute (in  
1431 Group 1) rather than including the document data itself. Before returning the response, the Printer  
1432 MUST validate that the Printer supports the retrieval method (e.g., http, ftp, etc.) implied by the URI,  
1433 and MUST check for valid URI syntax. If the client-supplied URI scheme is not supported, i.e. the value  
1434 is not in the Printer object's "referenced-uri-scheme-supported" attribute, the Printer object  
1435 **SHALLMUST** reject the request and return the 'client-error-uri-scheme-not-supported' status code. See  
1436 Section 16.3.5 for suggested additional checks. The Printer NEED NOT follow the reference and  
1437 validate the contents of the reference.

1438 If the Printer object supports this operation, it MUST support the "reference-uri-schemes-supported"  
1439 Printer attribute (see section 4.4.24).

1440 It is up to the IPP object to interpret the URI and subsequently "pull" the document from the source  
1441 referenced by the URI string.

### 1442 3.2.3 Validate-Job Operation

1443 This **MANDATORYREQUIRED** operation is similar to the Print-Job operation (section 3.2.1) except  
1444 that a client supplies no document data and the Printer allocates no resources (i.e., it does not create a  
1445 new Job object). This operation is used only to verify capabilities of a printer object against whatever  
1446 attributes are supplied by the client in the Validate-Job request. By using the Validate-Job operation a  
1447 client can validate that an identical Print-Job operation (with the document data) would be accepted. The  
1448 Validate-Job operation also performs the same security negotiation as the Print-Job operation (see section  
1449 8), so that a client can check that the client and Printer object security requirements can be met before  
1450 performing a Print-Job operation.

1451 Note: The Validate-Job operation does not accept a "document-uri" attribute in order to allow a client to  
1452 check that the same Print-URI operation will be accepted, since the client doesn't send the data with the  
1453 Print-URI operation. The client SHOULD just issue the Print-URI request.

1454 The Printer object returns the same status codes, Operation Attributes (Group 1) and Unsupported  
1455 Attributes (Group 2) as the Print-Job operation. However, no Job Object Attributes (Group 3) are  
1456 returned, since no Job object is created.

## 1457 3.2.4 Create-Job Operation

1458 This OPTIONAL operation is similar to the Print-Job operation (section 3.2.1) except that in the Create-  
1459 Job request, a client does not supply document data or any reference to document data. Also, the client  
1460 does not supply any of the "document-name", "document-format", "compression", or "document-natural-  
1461 language" operation attributes. This operation is followed by one or more Send-Documents or Send-URI  
1462 operations. In each of those operation requests, the client OPTIONALLY supplies the "document-  
1463 name", "document-format", and "document-natural-language" attributes for each document in the multi-  
1464 document Job object. If a Printer object supports the Create-Job operation, it MUST also support the  
1465 Send-Documents operation and also MAY support the Send-URI operation.

## 1466 3.2.5 Get-Printer-Attributes Operation

1467 This **MANDATORYREQUIRED** operation allows a client to request the values of the attributes of a  
1468 Printer object. In the request, the client supplies the set of Printer attribute names and/or attribute group  
1469 names in which the requester is interested. In the response, the Printer object returns a corresponding  
1470 attribute set with the appropriate attribute values filled in.

1471 For Printer objects, the possible names of attribute groups are:

- 1472 - 'job-template': all of the Job Template attributes that apply to a Printer object (the last two columns  
1473 of the table in Section 4.2).
- 1474 - 'printer-description': the attributes specified in Section 4.4.
- 1475 - 'all': the special group 'all' that includes all supported attributes.

1476  
1477 Since a client MAY request specific attributes or named groups, there is a potential that there is some  
1478 overlap. For example, if a client requests, 'printer-name' and 'all', the client is actually requesting the  
1479 "printer-name" attribute twice: once by naming it explicitly, and once by inclusion in the 'all' group. In  
1480 such cases, the Printer object NEED NOT return each attribute only once in the response even if it is  
1481 requested multiple times. The client SHOULD NOT request the same attribute in multiple ways.

1482 It is NOT REQUIRED that a Printer object support all attributes belonging to a group (since some  
1483 attributes are OPTIONAL). However, it is **MANDATORYREQUIRED** that each Printer object support  
1484 all group names.

## 1485 3.2.5.1 Get-Printer-Attributes Request

1486 The following sets of attributes are part of the Get-Printer-Attributes Request:

## 1487 Group 1: Operation Attributes

1488 Natural Language and Character Set:

1489 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1490 3.1.4.1.

1491

1492 Target:

1493 The "printer-uri" (uri) operation attribute which is the target for this operation as described in  
1494 section 3.1.5.

1495

1496 Requesting User Name:

1497 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1498 described in section 8.3.

1499

1500 "requested-attributes" (1setOf keyword) :

1501 The client OPTIONALLY supplies a set of attribute names and/or attribute group names in whose  
1502 values the requester is interested. The Printer object MUST support this attribute. If the client  
1503 omits this attribute, the Printer **SHALLMUST** respond as if this attribute had been supplied with a  
1504 value of 'all'.

1505

1506 "document-format" (mimeMediaType) :

1507 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1508 attribute. This attribute is useful for a Printer object to determine the set of supported attribute  
1509 values that relate to the requested document format. The Printer object **SHALLMUST** return the  
1510 attributes and values that it uses to validate a job on a create or Validate-Job operation in which  
1511 this document format is supplied. The Printer object SHOULD return only (1) those attributes  
1512 that are supported for the specified format and (2) the attribute values that are supported for the  
1513 specified document format. By specifying the document format, the client can get the Printer  
1514 object to eliminate the attributes and values that are not supported for a specific document format.  
1515 For example, a Printer object might have multiple interpreters to support both  
1516 'application/postscript' (for PostScript) and 'text/plain' (for text) documents. However, for only  
1517 one of those interpreters might the Printer object be able to support "number-up" with values of  
1518 '1', '2', and '4'. For the other interpreter it might be able to only support "number-up" with a value  
1519 of '1'. Thus a client can use the Get-Printer-Attributes operation to obtain the attributes and values  
1520 that will be used to accept/reject a create job operation.

1521

1522 **Note:**—If the Printer object does not distinguish between different sets of supported values for  
1523 each different document format when validating jobs in the create and Validate-Job operations, it  
1524 **SHALLMUST** NOT distinguish between different document formats in the Get-Printer-Attributes  
1525 operation. If the Printer object does distinguish between different sets of supported values for  
1526 each different document format specified by the client, this specialization applies only to the  
1527 following Printer object attributes:

- 1528  
1529           - Printer attributes that are Job Template attributes ("xxx-default" ~~and~~ "xxx"-supported", and  
1530           "xxx-ready" in the Table in Section 4.2),  
1531           - "pdl-override-supported",  
1532           - "compression-supported",  
1533           - "job-k-octets-supported",  
1534           - "job-impressions-supported",  
1535           - "job-media-sheets-supported"  
1536           - "printer-driver-installer",  
1537           - "color-supported", and  
1538           - "reference-uri-schemes-supported"

1539  
1540           The values of all other Printer object attributes (including "document-format-supported") remain  
1541           invariant with respect to the client supplied document format (except for new Printer description  
1542           attribute as registered according to section 6.2).

1543  
1544           If the client omits this "document-format" operation attribute, the Printer object ~~SHALL~~MUST  
1545           respond as if the attribute had been supplied with the value of the Printer object's "document-  
1546           format-default" attribute. It is recommended that the client always supply a value for "document-  
1547           format", since the Printer object's "document-format-default" may be 'application/octet-stream', in  
1548           which case the returned attributes and values are for the union of the document formats that the  
1549           Printer can automatically sense. For more details, see the description of the 'mimeType' attribute  
1550           syntax in section 4.1.9.

1551  
1552           If the client supplies a value for the "document-format" Operation attribute that is not supported  
1553           by the Printer, i.e., is not among the values of the Printer object's "document-format-supported"  
1554           attribute, the Printer object ~~SHALL~~MUST reject the operation and return the 'client-error-  
1555           document-format-not-supported' status code.  
1556

### 1557 3.2.5.2 Get-Printer-Attributes Response

1558           The Printer object returns the following sets of attributes as part of the Get-Printer-Attributes Response:

#### 1559 Group 1: Operation Attributes

##### 1560           Status Message:

1561           In addition to the ~~MANDATORY~~REQUIRED status code returned in every response, the  
1562           response ~~OPTIONALLY~~ includes a "status-message" (text) operation attribute as described in  
1563           section 3.1.5.  
1564

1565 Natural Language and Character Set:

1566 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1567 3.1.4.2.

1568

1569 Group 2: Unsupported Attributes

1570 This is a set of Operation attributes supplied by the client (in the request) that are not supported  
1571 by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16).

1572

1573 Group 3: Printer Object Attributes

1574 This is the set of requested attributes and their current values. The Printer object ignores (does  
1575 not respond with) any requested attribute which is not supported. The Printer object MAY  
1576 respond with a subset of the supported attributes and values, depending on the security policy in  
1577 force. However, the Printer object **SHALLMUST** respond with the 'unknown' value for any  
1578 supported attribute (including all **MANDATORYREQUIRED** attributes) for which the Printer  
1579 object does not know the value. Also the Printer object **SHALLMUST** respond with the 'no-  
1580 value' for any supported attribute (including all **MANDATORYREQUIRED** attributes) for which  
1581 the system administrator has not configured a value. See the description of the "out-of-band"  
1582 values in the beginning of Section 4.1.

1583

1584 3.2.6 Get-Jobs Operation

1585 This **MANDATORYREQUIRED** operation allows a client to retrieve the list of Job objects belonging to  
1586 the target Printer object. The client may also supply a list of Job attribute names and/or attribute group  
1587 names. A group of Job object attributes will be returned for each Job object that is returned.

1588 This operation is similar to the Get-Job-Attributes operation, except that this Get-Jobs operation returns  
1589 attributes from possibly more than one object (see the description of Job attribute group names in section  
1590 3.3.4).

1591 3.2.6.1 Get-Jobs Request

1592 The client submits the Get-Jobs request to a Printer object.

1593 The following groups of attributes are part of the Get-Jobs Request:

1594 Group 1: Operation Attributes

1595 Natural Language and Character Set:

1596 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1597 3.1.4.1.

1598

1599 Target:

1600 The "printer-uri" (uri) operation attribute which is the target for this operation as described in  
1601 section 3.1.5.

1602

1603 Requesting User Name:

1604 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1605 described in section 8.3.

1606

1607 "limit" (integer(1:MAX)):

1608 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1609 attribute. It is an integer value that indicates a limit to the number of Job objects returned. The  
1610 limit is a "stateless limit" in that if the value supplied by the client is 'N', then only the first 'N' jobs  
1611 are returned in the Get-Jobs Response. There is no mechanism to allow for the next 'M' jobs after  
1612 the first 'N' jobs. If the client does not supply this attribute, the Printer object responds with all  
1613 applicable jobs.

1614

1615 "requested-attributes" (1setOf keyword):

1616 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1617 attribute. It is a set of Job attribute names and/or attribute groups names in whose values the  
1618 requester is interested. This set of attributes is returned for each Job object that is returned. The  
1619 allowed attribute group names are the same as those defined in the Get-Job-Attributes operation  
1620 in section 3.3.4. If the client does not supply this attribute, the Printer **SHALLMUST** respond as  
1621 if the client had supplied this attribute with two values: 'job-uri' and 'job-id'.

1622

1623 "which-jobs" (keyword):

1624 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1625 attribute. It indicates which Job objects **SHALLMUST** be returned by the Printer object. The  
1626 values for this attribute are:

1627

1628 'completed': This includes any Job object whose state is 'completed', 'canceled', or 'aborted'.

1629 'not-completed': This includes any Job object whose state is 'pending', 'processing',

1630 'processing-stopped', or 'pending-held'.

1631

1632 A Printer object **SHALLMUST** support both values. However, if the implementation does not  
1633 keep jobs in the 'completed', 'canceled', and 'aborted' states, then it returns no jobs when the  
1634 'completed' value is supplied.

1635

1636 If a client supplies some other value, the Printer object **SHALLMUST** copy the attribute and the  
1637 unsupported value to the Unsupported Attributes response group, reject the request, and return  
1638 the 'client-error-attributes-or-values-not-supported' status code.

1639

1640 If the client does not supply this attribute, the Printer object **SHALLMUST** respond as if the  
1641 client had supplied the attribute with a value of 'not-completed'.

1642

1643 "my-jobs" (boolean):

1644 The client **OPTIONALLY** supplies this attribute. The Printer object **MUST** support this  
1645 attribute. It indicates whether all jobs or just the jobs submitted by the requesting user of this  
1646 request **SHALLMUST** be returned by the Printer object. If the client does not supply this  
1647 attribute, the Printer object **SHALLMUST** respond as if the client had supplied the attribute with  
1648 a value of 'false', i.e., all jobs. The means for authenticating the requesting user and matching the  
1649 jobs is described in section 8.

### 1650 3.2.6.2 Get-Jobs Response

1651 The Printer object returns all of the Job objects that match the criteria as defined by the attribute values  
1652 supplied by the client in the request. It is possible that no Job objects are returned since there may  
1653 literally be no Job objects at the Printer, or there may be no Job objects that match the criteria supplied by  
1654 the client. If the client requests any Job attributes at all, there is a set of Job Object Attributes returned  
1655 for each Job object.

#### 1656 Group 1: Operation Attributes

##### 1657 Status Message:

1658 In addition to the **MANDATORYREQUIRED** status code returned in every response, the  
1659 response **OPTIONALLY** includes a "status-message" (text) operation attribute as described in  
1660 section 3.1.5.

1661

##### 1662 Natural Language and Character Set:

1663 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1664 3.1.4.2.

1665

#### 1666 Group 2: Unsupported Attributes

1667 This is a set of Operation attributes supplied by the client (in the request) that are not supported  
1668 by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16.3).

1669

#### 1670 Groups 3 to N: Job Object Attributes



1671 The Printer object responds with one set of Job Object Attributes for each returned Job object.  
1672 The Printer object ignores (does not respond with) any requested attribute or value which is not  
1673 supported or which is restricted by the security policy in force, including whether the requesting  
1674 user is the user that submitted the job (job originating user) or not (see section 8). However, the  
1675 Printer object **SHALLMUST** respond with the 'unknown' value for any supported attribute  
1676 (including all **MANDATORYREQUIRED** attributes) for which the Printer object does not know  
1677 the value, unless it would violate the security policy. See the description of the "out-of-band"  
1678 values in the beginning of Section 4.1.

1679  
1680 For any job submitted in a different natural language than the natural language that the Printer  
1681 object is returning in the "attributes-natural-language" operation attribute in the Get-Jobs  
1682 response, the Printer **SHALLMUST** indicate the submitted natural language by returning the Job  
1683 object's "attributes-natural-language" as the first Job object attribute, which overrides the  
1684 "attributes-natural-language" operation attribute value being returned by the Printer object. If any  
1685 returned 'text' or 'name' attribute includes a Natural Language Override as described in the  
1686 sections 4.1.1.2 and 4.1.2.2, the Natural Language Override overrides the Job object's "attributes-  
1687 natural-language" value and/or the "attributes-natural-language" operation attribute value.

1688  
1689 Jobs are returned in the following order:

- 1690 - If the client requests all 'completed' Jobs (Jobs in the 'completed', 'aborted', or 'canceled'  
1691 states), then the Jobs are returned newest to oldest (with respect to actual completion  
1692 time)
  - 1693 - If the client requests all 'not-completed' Jobs (Jobs in the 'pending', 'processing', 'pending-  
1694 held', and 'processing-stopped' states), then Jobs are returned in relative chronological  
1695 order of expected time to complete (based on whatever scheduling algorithm is configured  
1696 for the Printer object).
- 1697

### 1698 3.3 Job Operations

1699 All Job operations are directed at Job objects. A client **MUST** always supply some means of identifying  
1700 the Job object in order to identify the correct target of the operation. That job identification **MAY** either  
1701 be a single Job URI or a combination of a Printer URI with a Job ID. The IPP object implementation  
1702 **MUST** support both forms of identification for every job.

#### 1703 3.3.1 Send-Document Operation

1704 This **OPTIONAL** operation allows a client to create a multi-document Job object that is initially "empty"  
1705 (contains no documents). In the Create-Job response, the Printer object returns the Job object's URI (the  
1706 "job-uri" attribute) and the Job object's 32-bit identifier (the "job-id" attribute). For each new document



1707 that the client desires to add, the client uses a Send-Document operation. Each Send-Document Request  
1708 contains the entire stream of document data for one document.

1709 Since the Create-Job and the send operations (Send-Document or Send-URI operations) that follow can  
1710 occur over arbitrarily long periods of time, each Printer object must decide how long to "wait" for the  
1711 next send operation. The Printer object OPTIONALLY supports the "multiple-operation-timeout"  
1712 attribute. This attribute indicates the maximum number of seconds the Printer object will wait for the  
1713 next send operation. If the Printer object times-out waiting for the next send operation, the Printer object  
1714 MAY decide on any of the following semantic actions:

- 1715 1. Assume that the Job is an invalid job, start the process of changing the job state to 'aborted', and  
1716 clean up all resources associated with the Job. In this case, if another send operation is finally  
1717 received, the Printer responds with a "client-error-not-possible" or "client-error-not-found"  
1718 depending on whether or not the Job object is still around when it finally arrives.
- 1719 2. Assume that the last send operation received was in fact the last document (as if the "last-  
1720 document" flag had been set to 'true'), close the Job object, and proceed to process it (i.e., move  
1721 the Job's state to 'pending').
- 1722 3. Assume that the last send operation received was in fact the last document, close the Job, but move  
1723 it to the 'pending-held' to allow an operator to determine whether or not to continue processing  
1724 the Job by moving it back to the 'pending' state.

1725  
1726 Each implementation is free to decide the "best" action to take depending on local policy, the value of  
1727 "ipp-attribute-fidelity", and/or any other piece of information available to it. If the choice is to abort the  
1728 Job object, it is possible that the Job object may already have been processed to the point that some  
1729 media sheet pages have been printed.

### 1730 3.3.1.1 Send-Document Request

1731 The following attribute sets are part of the Send-Document Request:

#### 1732 Group 1: Operation Attributes

##### 1733 Natural Language and Character Set:

1734 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1735 3.1.4.1.

1736

##### 1737 Target:

1738 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation  
1739 attribute(s) which define the target for this operation as described in section 3.1.5.

1740

1741 Requesting User Name:

1742 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1743 described in section 8.3.

1744

1745 "document-name" (name(MAX)):

1746 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1747 attribute. It contains the client supplied document name. The document name MAY be different  
1748 than the Job name. It might be helpful, but NEED NOT be unique across multiple documents in  
1749 the same Job. Typically, the client software automatically supplies the document name on behalf  
1750 of the end user by using a file name or an application generated name. See the description of the  
1751 "document-name" operation attribute in the Print-Job Request (section 3.2.1.1) for more  
1752 information about this attribute.

1753

1754 "document-format" (mimeMediaType) :

1755 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1756 attribute. The value of this attribute identifies the format of the supplied document data. If the  
1757 client does not supply this attribute, the Printer object assumes that the document data is in the  
1758 format defined by the Printer object's "document-format-default" attribute. If the client supplies  
1759 this attribute, but the value is not supported by the Printer object, i.e., the value is not one of the  
1760 values of the Printer object's "document-format-supported" attribute, the Printer object  
1761 **SHALLMUST** reject the request and return the 'client-error-document-format-not-supported'  
1762 status code.

1763

1764 "document-natural-language" (naturalLanguage):

1765 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports  
1766 this attribute. This attribute specifies the natural language of the document for those document-  
1767 formats that require a specification of the natural language in order to image the document  
1768 unambiguously. There are no particular values required for the Printer object to support.

1769

1770 "compression" (type3 keyword)

1771 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports  
1772 this attribute and the "compression-supported" attribute (see section 4.4.29). The client supplied  
1773 "compression" operation attribute identifies the compression algorithm used on the document  
1774 data. If the client omits this attribute, the Printer object **SHALLMUST** assume that the data is  
1775 not compressed. If the client supplies the attribute and the Printer object supports the attribute,  
1776 the Printer object **SHALLMUST** use the corresponding decompression algorithm on the  
1777 document data. If the client supplies this attribute, but the value is not supported by the Printer  
1778 object, i.e., the value is not one of the values of the Printer object's "compression-supported"  
1779 attribute, the Printer object **SHALLMUST** copy the attribute and its value to the Unsupported

1780 Attributes response group, reject the request, and return the 'client-error-attributes-or-values-not-  
1781 supported' status code.

1782  
1783 "last-document" (boolean):

1784 The client MUST supply this attribute. The Printer object MUST support this attribute. It is a  
1785 boolean flag that is set to 'true' if this is the last document for the Job, 'false' otherwise.

1786  
1787 Group 2: Document Content

1788 The client MUST supply the document data if the "last-document" flag is set to 'false'. However,  
1789 since a client might not know that the previous document sent with a Send-Document (or Send-  
1790 URI) operation was the last document (i.e., the "last-document" attribute was set to 'false'), it is  
1791 legal to send a Send-Document request with no document data where the "last-document" flag is  
1792 set to 'true'. Such a request ~~SHALL~~MUST NOT increment the value of the Job object's "number-  
1793 of-documents" attribute, since no real document was added to the job.

1794 3.3.1.2 Send-Document Response

1795 The following sets of attributes are part of the Send-Document Response:

1796 Group 1: Operation Attributes

1797 Status Message:

1798 In addition to the ~~MANDATORY~~REQUIRED status code returned in every response, the  
1799 response OPTIONALLY includes a "status-message" (text) operation attribute as described in  
1800 section 3.1.5.

1801  
1802 Natural Language and Character Set:

1803 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1804 3.1.4.2.

1805  
1806 Group 2: Unsupported Attributes

1807 This is a set of Operation attributes supplied by the client (in the request) that are not supported  
1808 by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16.3).

1809  
1810 Group 3: Job Object Attributes

1811 This is the same set of attributes as described in the Print-Job response (see section 3.2.1.2).

1812

## 1813 3.3.2 Send-URI Operation

1814 This OPTIONAL operation is identical to the Send-Document operation (see section 3.3.1) except that a  
1815 client MUST supply a URI reference ("document-uri" operation attribute) rather than the document data  
1816 itself. If a Printer object supports this operation, clients can use both Send-URI or Send-Document  
1817 operations to add new documents to an existing multi-document Job object. However, if a client needs  
1818 to indicate that the previous Send-URI or Send-Document was the last document, the client MUST use  
1819 the Send-Document operation with no document data and the "last-document" flag set to 'true' (rather  
1820 than using a Send-URI operation with no "document-uri" operation attribute). If a Printer object  
1821 supports this operation, it MUST also support the Print-URI operation (see section 3.2.2).

1822 The Printer object MUST validate the syntax and URI scheme of the supplied URI before returning a  
1823 response, just as in the Print-URI operation.

## 1824 3.3.3 Cancel-Job Operation

1825 This **MANDATORYREQUIRED** operation allows a client to cancel a Print Job any time after a create  
1826 job operation. Since a Job might already be printing by the time a Cancel-Job is received, some media  
1827 sheet pages might be printed before the job is actually terminated.

## 1828 3.3.3.1 Cancel-Job Request

1829 The following groups of attributes are part of the Cancel-Job Request:

## 1830 Group 1: Operation Attributes

1831 Natural Language and Character Set:

1832 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1833 3.1.4.1.

1834

1835 Target:

1836 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation  
1837 attribute(s) which define the target for this operation as described in section 3.1.5.

1838

1839 Requesting User Name:

1840 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1841 described in section 8.3.

1842

1843 "message" (text(127)):

1844 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports  
1845 this attribute. It is a message to the operator. This "message" attribute is not the same as the "job-

1846 message-from-operator" attribute. That attribute is used to report a message from the operator to  
1847 the end user that queries that attribute. This "message" operation attribute is used to send a  
1848 message from the client to the operator along with the operation request. It is an implementation  
1849 decision of how or where to display this message to the operator (if at all).  
1850

### 1851 3.3.3.2 Cancel-Job Response

1852 The following sets of attributes are part of the Cancel-Job Response:

#### 1853 Group 1: Operation Attributes

##### 1854 Status Message:

1855 In addition to the **MANDATORYREQUIRED** status code returned in every response, the  
1856 response **OPTIONALLY** includes a "status-message" (text) operation attribute as described in  
1857 section 3.1.5.

1858  
1859 If the job is already in the 'completed', 'aborted', or 'canceled' state, or the 'process-to-stop-point'  
1860 value is set in the Job's "job-state-reasons" attribute, the Printer object **SHALLMUST** reject the  
1861 request and return the 'client-error-not-possible' error status code.  
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.

#### 1866 Group 2: Unsupported Attributes

1868 This is a set of Operation attributes supplied by the client (in the request) that are not supported  
1869 by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16.3).  
1870

1871 Once a successful response has been sent, the implementation guarantees that the Job will eventually end  
1872 up in the 'canceled' state. Between the time of the Cancel-Job operation is accepted and when the job  
1873 enters the 'canceled' job-state (see section 4.3.7), the "job-state-reasons" attribute **SHOULD** contain the '  
1874 processing-to-stop-point' value which indicates to later queries that although the Job might still be  
1875 'processing', it will eventually end up in the 'canceled' state, not the 'completed' state.

### 1876 3.3.4 Get-Job-Attributes Operation

1877 This **MANDATORYREQUIRED** operation allows a client to request the values of attributes of a Job  
1878 object and it is almost identical to the Get-Printer-Attributes operation (see section 3.2.5). The only

1879 differences are that the operation is directed at a Job object rather than a Printer object, there is no  
1880 "document-format" operation attribute used when querying a Job object, and the returned attribute group  
1881 is a set of Job object attributes rather than a set of Printer object attributes.

1882 For Jobs, the possible names of attribute groups are:

- 1883 - 'job-template': all of the Job Template attributes that apply to a Job object (the first column of the  
1884 table in Section 4.2).
- 1885 - 'job-description': all of the Job Description attributes specified in Section 4.3.
- 1886 - 'all': the special group 'all' that includes all supported attributes.

1887

1888 Since a client MAY request specific attributes or named groups, there is a potential that there is some  
1889 overlap. For example, if a client requests, 'job-name' and 'job-description', the client is actually requesting  
1890 the "job-name" attribute once by naming it explicitly, and once by inclusion in the 'job-description' group.  
1891 In such cases, the Printer object NEED NOT return the attribute only once in the response even if it is  
1892 requested multiple times. The client SHOULD NOT request the same attribute in multiple ways.

1893 It is NOT REQUIRED that a Job object support all attributes belonging to a group (since some attributes  
1894 are OPTIONAL). However it is **MANDATORYREQUIRED** that each Job object support all group  
1895 names.

#### 1896 3.3.4.1 Get-Job-Attributes Request

1897 The following groups of attributes are part of the Get-Job-Attributes Request when the request is  
1898 directed at a Job object:

##### 1899 Group 1: Operation Attributes

###### 1900 Natural Language and Character Set:

1901 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1902 3.1.4.1.

1903

###### 1904 Target:

1905 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation  
1906 attribute(s) which define the target for this operation as described in section 3.1.5.

1907

###### 1908 Requesting User Name:

1909 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1910 described in section 8.3.

1911

1912 "requested-attributes" (1setOf keyword) :

1913 The client **OPTIONALLY** supplies this attribute. The IPP object **MUST** support this attribute.  
1914 It is a set of attribute names and/or attribute group names in whose values the requester is  
1915 interested. If the client omits this attribute, the IPP object **SHALLMUST** respond as if this  
1916 attribute had been supplied with a value of 'all'.  
1917

### 1918 3.3.4.2 Get-Job-Attributes Response

1919 The Printer object returns the following sets of attributes as part of the Get-Job-Attributes Response:

#### 1920 Group 1: Operation Attributes

##### 1921 Status Message:

1922 In addition to the **MANDATORYREQUIRED** status code returned in every response, the  
1923 response **OPTIONALLY** includes a "status-message" (text) operation attribute as described in  
1924 section 3.1.5.  
1925

##### 1926 Natural Language and Character Set:

1927 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1928 3.1.4.2. The "attributes-natural-language" **MAY** be the natural language of the Job object, rather  
1929 than the one requested.  
1930

#### 1931 Group 2: Unsupported Attributes

1932 This is a set of Operation attributes supplied by the client (in the request) that are not supported  
1933 by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16.3).  
1934

#### 1935 Group 3: Job Object Attributes

1936 This is the set of requested attributes and their current values. The IPP object ignores (does not  
1937 respond with) any requested attribute or value which is not supported or which is restricted by the  
1938 security policy in force, including whether the requesting user is the user that submitted the job  
1939 (job originating user) or not (see section 8). However, the IPP object **SHALLMUST** respond  
1940 with the 'unknown' value for any supported attribute (including all **MANDATORYREQUIRED**  
1941 attributes) for which the IPP object does not know the value, unless it would violate the security  
1942 policy. See the description of the "out-of-band" values in the beginning of Section 4.1.

## 1943 4. Object Attributes

1944 This section describes the attributes with their corresponding attribute syntaxes and values that are part of  
1945 the IPP model. The sections below show the objects and their associated attributes which are included  
1946 within the scope of this protocol. Many of these attributes are derived from other relevant specifications:

- 1947 - Document Printing Application (DPA) [ISO10175]
- 1948 - RFC 1759 Printer MIB [RFC1759]

1949

1950 Each attribute is uniquely identified in this document using a "keyword" (see section 13.2.1) which is the  
1951 name of the attribute. The keyword is included in the section header describing that attribute.

1952 Note: Not only are keywords used to identify attributes, but one of the attribute syntaxes described  
1953 below is "keyword" so that some attributes have keyword values. Therefore, these attributes are defined  
1954 as having an attribute syntax that is a set of keywords.

## 1955 4.1 Attribute Syntaxes

1956 This section defines the basic attribute syntax types that all clients and IPP objects **SHALLMUST** be able  
1957 to accept in responses and accept in requests, respectively. Each attribute description in sections 3 and 4  
1958 includes the name of attribute syntax(es) in the heading (in parentheses). A conforming implementation  
1959 of an attribute **SHALLMUST** include the semantics of the attribute syntax(es) so identified. Section 6.3  
1960 describes how the protocol can be extended with new attribute syntaxes.

1961 The attribute syntaxes are specified in the following sub-sections, where the sub-section heading is the  
1962 keyword name of the attribute syntax inside the single quotes. In operation requests and responses each  
1963 attribute value **MUST** be represented as one of the attribute syntaxes specified in the sub-section heading  
1964 for the attribute. In addition, the value of an attribute in a response (but not in a request) **MAY** be one of  
1965 the "out-of-band" values. Standard "out-of-band" values are:

1966 'unknown': The attribute is supported by the IPP object, but the value is unknown to the IPP object  
1967 for some reason.

1968 'unsupported': The attribute is unsupported by the IPP object. This value **SHALLMUST** be returned  
1969 only as the value of an attribute in the Unsupported Attributes Group.

1970 'no-value': The attribute is supported by the Printer object, but the system administrator has not yet  
1971 configured a value.

1972

1973 The protocol specification defines mechanisms for passing "out-of-band" values. All attributes in a  
1974 request **SHALLMUST** have one or more values as defined in Sections 4.2 to 4.4. Thus clients



1975 ~~SHALLMUST~~ NOT supply attributes with "out-of-band" values. All attribute in a response  
1976 ~~SHALLMUST~~ have one or more values as defined in Sections 4.2 to 4.4 or a single "out-of-band" value.

1977 Most attributes are defined to have a single attribute syntax. However, a few attributes (e.g., "job-sheet",  
1978 "media", "job-hold-until") are defined to have several attribute syntaxes, depending on the value. These  
1979 multiple attribute syntaxes are separated by the "|" character in the sub-section heading to indicate the  
1980 choice. Since each value ~~SHALLMUST~~ be tagged as to its attribute syntax in the protocol, a single-  
1981 valued attribute instance may have any one of its attribute syntaxes and a multi-valued attribute instance  
1982 may have a mixture of its defined attribute syntaxes.

#### 1983 4.1.1 'text'

1984 A text attribute is an attribute whose value is a sequence of zero or more characters encoded in a  
1985 maximum of 1023 ('MAX') octets. MAX is the maximum length for ~~all each~~ values of any text attribute.  
1986 However, if an attribute will always contain values whose maximum length is much less than MAX, the  
1987 definition of that attribute will include a qualifier that defines the maximum length for values of that  
1988 attribute. For example: the "printer-location" attribute is specified as "printer-location (text(127))". In  
1989 this case, text values for "printer-location" ~~SHALLMUST~~ NOT exceed 127 octets; if supplied with a  
1990 longer text string via some external interface (other than the protocol), implementations are free to  
1991 truncate to this shorter length limitation.

1992 In this specification, all text attributes are defined using the 'text' syntax. However, 'text' is used only for  
1993 brevity; the formal interpretation of 'text' is: 'textWithoutLanguage | textWithLanguage'. That is, for any  
1994 attribute defined in this specification using the 'text' attribute syntax, all IPP objects and clients  
1995 ~~SHALLMUST~~ ~~accept, support both, and return either~~ the 'textWithoutLanguage' ~~or and~~  
1996 'textWithLanguage' attribute syntaxes. However, in actual usage and protocol execution, objects and  
1997 clients accept and return only one of the two syntax per attribute. The syntax 'text' never appears "on-  
1998 the-wire".

1999 Both 'textWithoutLanguage' and 'textWithLanguage' are needed to support the real world needs of  
2000 interoperability between sites and systems that use different natural languages as the basis for human  
2001 communication. Generally, one natural language applies to all text attributes in a given request or  
2002 response. The language is indicated by the "attributes-natural-language" operation attribute defined in  
2003 section 3.1.4 or "attributes-natural-language" job attribute defined in section 4.3.24, and there is no need  
2004 to identify the natural language for each text string on a value-by-value basis. In these cases, the attribute  
2005 syntax 'textWithoutLanguage' is used for text attributes. In other cases, the client needs to supply or the  
2006 Printer object needs to return a text value in a natural language that is different from the rest of the text  
2007 values in the request or response. In these cases, the client or Printer object uses the attribute syntax  
2008 'textWithLanguage' for text attributes (this is the Natural Language Override mechanism described in  
2009 section 3.1.4).

2010 **The** 'textWithoutLanguage' and 'textWithLanguage' **attribute syntaxes** are described in more detail in the  
2011 following sections.

#### 2012 4.1.1.1 'textWithoutLanguage'

2013 The 'textWithoutLanguage' syntax indicates a value that is sequence of zero or more characters. Text  
2014 strings are encoded using the rules of some charset. The Printer object **SHALLMUST** support the UTF-  
2015 8 charset [RFC2044] and MAY support additional charsets to represent 'text' values, provided that the  
2016 charsets are registered with IANA [IANA-CS]. See Section 4.1.7 for the specification of the 'charset'  
2017 attribute syntax, including restricted semantics and examples of charsets.

#### 2018 4.1.1.2 'textWithLanguage'

2019 The 'textWithLanguage' attribute syntax is a compound attribute syntax consisting of two parts: a  
2020 'textWithoutLanguage' part plus an additional 'naturalLanguage' (see section 4.1.8) part that overrides the  
2021 natural language in force. The 'naturalLanguage' part explicitly identifies the natural language that applies  
2022 to the text part of that value and that value alone. For any give text attribute, the 'textWithoutLanguage'  
2023 part is limited to the maximum length defined for that attribute, , but the 'naturalLanguage' part is always  
2024 limited to 63 octets. Using the 'textWithLanguage' attribute syntax rather than the normal  
2025 'textWithoutLanguage' syntax is the so-called Natural Language Override mechanism and **MUST** be  
2026 supported by all IPP objects and clients.

2027

2028 If the attribute is multi-valued (1setOf text), then the 'textWithLanguage' attribute syntax **MUST** be used  
2029 to explicitly specify each attribute value whose natural language needs to be overridden. Other values in  
2030 a multi-valued 'text' attribute in a request or a response revert to the natural language of the operation  
2031 attribute or to the "attributes-natural-language" Job attribute, if present, in the case of a Get-Jobs  
2032 response.

2033

2034 In a create request, the Printer object **MUST** accept and store with the Job object any natural language in  
2035 the "attributes-natural-language" operation attribute, whether the Printer object supports that natural  
2036 language or not. Furthermore, the Printer object **MUST** accept and store any 'textWithLanguage'  
2037 attribute value, whether the Printer object supports that natural language or not. These requirements are  
2038 independent of the value of the "ipp-attribute-fidelity" operation attribute that the client MAY supply.

2039 Example: If the client supplies the "attributes-natural-language" operation attribute with the value: 'en'  
2040 indicating English, but the value of the "job-name" attribute is in French, the client **MUST** use the  
2041 'textWithLanguage' attribute syntax with the following two values:

2042 'fr': Natural Language Override indicating French  
2043 'Rapport Mensuel': the job name in French

2044  
2045 See the Protocol document [IPP-PRO] for a detailed example of the 'textWithLanguage' attribute syntax.

#### 2046 4.1.2 'name'

2047 This syntax type is used for user-friendly strings, such as a Printer name, that, for humans, are more  
2048 meaningful than identifiers. Names are usually never translated from one natural language to another.  
2049 The 'name' attribute syntax is essentially the same as 'text', including the **MANDATORYREQUIRED**  
2050 support of UTF-8 except that the sequence of characters is limited so that its encoded form  
2051 **SHALLMUST** NOT exceed 255 **(MAX)** octets.

2052  
2053 Also like 'text', 'name' is really an abbreviated notation for either 'nameWithoutLanguage' or  
2054 'nameWithLanguage'. That is, ; all IPP objects and clients MUST support both the  
2055 'nameWithoutLanguage' and 'nameWithLanguage' attribute syntaxes. However, in actual usage and  
2056 protocol execution, objects and clients accept and return only one of the two syntax per attribute. The  
2057 syntax 'name' never appears "on-the-wire". all IPP objects and clients MUST support the notion of 'name'  
2058 attributes using either the 'nameWithoutLanguage' or the 'nameWithLanguage' syntax during protocol  
2059 execution.

2060 Note: Only the 'text' and 'name' attribute syntaxes permit the Natural Language Override mechanism.

2061 Some attributes are defined as 'type3 keyword | name'. These attributes support values that are either  
2062 type3 keywords or names. This dual-syntax mechanism enables a site administrator to extend these  
2063 attributes to legally include values that are locally defined by the site administrator. Such names are not  
2064 registered with IANA.

##### 2065 4.1.2.1 'nameWithoutLanguage'

2066 The 'nameWithoutLanguage' syntax indicates a value that is sequence of zero or more characters so that  
2067 its encoded form does not exceed **127-MAX** octets.

##### 2068 4.1.2.2 'nameWithLanguage'

2069 The 'nameWithLanguage' attribute syntax is a compound attribute syntax consisting of two parts: a  
2070 'nameWithoutLanguage' part plus an additional 'naturalLanguage' (see section 4.1.8) part that overrides  
2071 the natural language in force. The 'naturalLanguage' part explicitly identifies the natural language that  
2072 applies to that name value and that name value alone.

2073 The 'nameWithLanguage' attribute syntax behaves ~~that-the~~ same as the 'textWithLanguage' syntax. If a  
2074 name is in a language that is different than the rest of the object or operation, then this  
2075 'nameWithLanguage' syntax is used rather than the generic 'nameWithoutLanguage' syntax.

2076 ~~The 'nameWithLanguage' attribute syntax is a compound attribute syntax consisting of two parts: a~~  
2077 ~~'nameWithoutLanguage' part plus an additional 'naturalLanguage' (see section 4.1.8) part that overrides~~  
2078 ~~the natural language in force. The 'naturalLanguage' part explicitly identifies the natural language that~~  
2079 ~~applies to the that name value and that name value alone.~~

2080 Example: If the client supplies the "attributes-natural-language" operation attribute with the value: 'en'  
2081 indicating English, but the "printer-name" attribute is in German, the client MUST use the  
2082 'nameWithLanguage' attribute syntax as follows:

2083       'de': Natural Language Override indicating German

2084       'Farbdrucker': the Printer name in German

2085

#### 2086 4.1.3 'keyword'

2087 The 'keyword' attribute syntax is a sequence of characters, length: 1 to 255, containing only the US-  
2088 ASCII [ASCII] encoded values for lowercase letters ("a" - "z"), digits ("0" - "9"), hyphen ("-"), dot ("."),  
2089 and underscore ("\_"). The first character MUST be a lowercase letter. Furthermore, keywords  
2090 ~~SHALLMUST~~ be in U.S. English.

2091 This syntax type is used for enumerating semantic identifiers of entities in the abstract protocol, i.e.,  
2092 entities identified in this document. Keywords are used as attribute names or values of attributes. Unlike  
2093 'text' and 'name' attribute values, 'keyword' values ~~SHALLMUST~~ NOT use the Natural Language  
2094 Override mechanism, since they ~~SHALLMUST~~ always be US-ASCII and U.S. English.

2095 Keywords are for use in the protocol. A user interface will likely provide a mapping between protocol  
2096 keywords and displayable user-friendly words and phrases which are localized to the natural language of  
2097 the user. While the keywords specified in this document MAY be displayed to users whose natural  
2098 language is U.S. English, they MAY be mapped to other U.S. English words for U.S. English users, since  
2099 the user interface is outside the scope of this document.

2100 In the definition for each attribute of this syntax type, the full set of defined keyword values for that  
2101 attribute are listed.

2102 When a keyword is used to represent an attribute (its name), it MUST be unique within the full scope of  
2103 all IPP objects and attributes. When a keyword is used to represent a value of an attribute, it MUST be  
2104 unique just within the scope of that attribute. That is, the same keyword ~~SHALLMUST~~ NOT be used for  
2105 two different values within the same attribute to mean two different semantic ideas. However, the same

2106 keyword MAY be used across two or more attributes, representing different semantic ideas for each  
2107 attribute. Section 6.1 describes how the protocol can be extended with new keyword values. Examples  
2108 of attribute name keywords:

2109 "job-name"  
2110 "attributes-charset"

2111  
2112 Note: This document uses "type1", "type2", and "type3" prefixes to the "keyword" ~~and "enum"~~ basic  
2113 syntaxes to indicate different levels of review for extensions (see section 6.1). ~~This extra information applies only to~~  
2114 ~~how the set of values defined for attributes with these syntaxes can be extended; this extra information is~~  
2115 ~~not carried in the protocol itself. "type1" indicates that new versions of the IPP standards documents~~  
2116 ~~must be revised and issued in order for new values to be added. "type2" indicates that IPP Subject~~  
2117 ~~Matter Experts must work with IANA to review and approve any proposed new values before the new~~  
2118 ~~values can be registered. "type3" indicates that IPP Subject Matter Experts are not required to review~~  
2119 ~~and approve any proposed new values before the new values can be registered with IANA. These~~  
2120 ~~extensibility mechanisms and restrictions are fully described in section 6.1-).~~

#### 2121 4.1.4 'enum'

2122 The 'enum' attribute syntax is an enumerated integer value that is in the range from 1 to 2\*\*31 - 1  
2123 (MAX). Each value has an associated 'keyword' name. In the definition for each attribute of this syntax  
2124 type, the full set of possible values for that attribute are listed. This syntax type is used for attributes for  
2125 which there are enum values assigned by other standards, such as SNMP MIBs. A number of attribute  
2126 enum values in this specification are also used for corresponding attributes in other standards [RFC1759].  
2127 This syntax type is not used for attributes to which the system administrator may assign values. Section  
2128 6.1 describes how the protocol can be extended with new enum values.

2129 Enum values are for use in the protocol. A user interface will provide a mapping between protocol enum  
2130 values and displayable user-friendly words and phrases which are localized to the natural language of the  
2131 user. While the enum symbols specified in this document MAY be displayed to users whose natural  
2132 language is U.S. English, they MAY be mapped to other U.S. English words for U.S. English users, since  
2133 the user interface is outside the scope of this document.

2134 Note: SNMP MIBs use '2' for 'unknown' which corresponds to the IPP "out-of-band" value 'unknown'.  
2135 See the description of the "out-of-band" values at the beginning of Section 4.1. Therefore, attributes of  
2136 type 'enum' start at '3'.

2137 Note: This document uses "type1", "type2", and "type3" prefixes to the "enum" basic syntax to indicate  
2138 different levels of review for extensions (see section 6.1).

## 2139 4.1.5 'uri'

2140 The 'uri' attribute syntax is any valid Uniform Resource Identifier or URI [RFC1630]. Most often, URIs  
2141 are simply Uniform Resource Locators or URLs [RFC1738] [RFC1808]. The maximum length of URIs  
2142 used within IPP is 1023 octets. Although most other IPP syntax types allow for only lower-cased values,  
2143 this syntax type allows for mixed-case values. The URI and URL standards allow for mixed-case values  
2144 that are case-sensitive.

## 2145 4.1.6 'uriScheme'

2146 The 'uriScheme' attribute syntax is a sequence of characters representing a URI scheme according to RFC  
2147 1738 [RFC1738]. Though RFC 1736 requires that the values be case-insensitive, IPP requires all lower  
2148 case to simplify comparing by IPP clients and Printer objects. Standard values for this syntax type are the  
2149 following keywords:

2150 'http': for HTTP schemed URIs (e.g., "http:...")

2151 'https': for use with ~~non-standard~~ HTTPS schemed URIs (e.g., "https:...") (not on standards track)

2152 'ftp': for FTP schemed URIs (e.g., "ftp:...")

2153 'mailto': for SMTP schemed URIs (e.g., "mailto:...")

2154 'file': for file schemed URIs (e.g., "file:...")

2155

2156 A Printer object MAY support any URI scheme that has been registered with IANA [IANA-MT]. The  
2157 maximum length of URI schemes used within IPP is 63 octets.

## 2158 4.1.7 'charset'

2159 The 'charset' attribute syntax is a standard identifier for a charset. A charset is a coded character set and  
2160 encoding scheme. Charsets are used for labeling certain document contents and 'text' and 'name' attribute  
2161 values. The syntax and semantics of this attribute syntax are specified in RFC 2046 [RFC2046] and  
2162 contained in the IANA character-set Registry [IANA-CS] according to the IANA procedures [~~IANA-~~  
2163 ~~CSa~~RFC2278]. Though RFC 2046 requires that the values be case-insensitive US-ASCII, IPP requires  
2164 all lower case to simplify comparing by IPP clients and Printer objects. When a character-set in the  
2165 IANA registry has more than one name (alias), the name labeled as "(preferred MIME name)", if present,  
2166 SHALLMUST be used.

2167 The maximum length of charset values used within IPP is 63 octets.

2168 Some examples are:

2169 'utf-8': ISO 10646 Universal Multiple-Octet Coded Character Set (UCS) represented as the UTF-8  
2170 [RFC2279] transfer encoding scheme in which US-ASCII is a subset charset.

2171 'us-ascii': 7-bit American Standard Code for Information Interchange (ASCII), ANSI X3.4-1986  
2172 [ASCII]. That standard defines US-ASCII, but RFC 2045 [46] eliminates most of the control  
2173 characters from conformant usage in MIME and IPP.  
2174 'iso-8859-1': 8-bit One-Byte Coded Character Set, Latin Alphabet Nr 1 [ISO8859-1]. That standard  
2175 defines a coded character set that is used by Latin languages in the Western Hemisphere and  
2176 Western Europe. US-ASCII is a subset charset.  
2177 'iso-10646-ucs-2': ISO 10646 Universal Multiple-Octet Coded Character Set (UCS) represented as  
2178 two octets (UCS-2), with the high order octet of each pair coming first (so-called Big Endian  
2179 integer).

2180

2181 Some attribute descriptions MAY place additional requirements on charset values that may be used, such  
2182 as ~~MANDATORYREQUIRED~~ values that MUST be supported or additional restrictions, such as  
2183 requiring that the charset have US-ASCII as a subset charset.

#### 2184 4.1.8 'naturalLanguage'

2185 The 'naturalLanguage' attribute syntax is a standard identifier for a natural language and optionally a  
2186 country. The values for this syntax type are defined by RFC 1766 [RFC1766]. Though RFC 1766  
2187 requires that the values be case-insensitive US-ASCII, IPP requires all lower case to simplify comparing  
2188 by IPP clients and Printer objects. Examples include:

2189 'en': for English  
2190 'en-us': for US English  
2191 'fr': for French  
2192 'de': for German

2193

2194 The maximum length of naturalLanguage values used within IPP is 63 octets.

#### 2195 4.1.9 'mimeMediaType'

2196 The 'mimeMediaType' attribute syntax is the Internet Media Type (sometimes called MIME type) as  
2197 defined by RFC 2046 [RFC2046] and registered according to the procedures of RFC 2048 [RFC2048]  
2198 for identifying a document format. The value MAY include a charset parameter, depending on the  
2199 specification of the Media Type in the IANA Registry [IANA-MT]. Although most other IPP syntax  
2200 types allow for only lower-cased values, this syntax type allows for mixed-case values which are case-  
2201 insensitive.

2202 Examples are:

2203 'text/html': An HTML document



2204 'text/plain': A plain text document in US-ASCII (RFC 2046 indicates that in the absence of the  
2205 charset parameter **SHALLMUST** mean US-ASCII rather than simply unspecified) [RFC2046].  
2206 'text/plain; charset=US-ASCII': A plain text document in US-ASCII [52, 56].  
2207 'text/plain; charset=ISO-8859-1': A plain text document in ISO 8859-1 (Latin 1) [ISO8859-1].  
2208 'text/plain; charset=utf-8': A plain text document in ISO 10646 represented as UTF-8 [RFC2044]  
2209 'text/plain, charset=iso-10646-ucs-2': A plain text document in ISO 10646 represented in two octets  
2210 (UCS-2) [ISO10646-1]  
2211 'application/postscript': A PostScript document [RFC2046]  
2212 'application/vnd.hp-PCL': A PCL document [IANA-MT] (charset escape sequence embedded in the  
2213 document data)  
2214 'application/octet-stream': Auto-sense - see below  
2215

2216 One special type is 'application/octet-stream'. If the Printer object supports this value, the Printer object  
2217 **SHALLMUST** be capable of auto-sensing the format of the document data. If the Printer object's default  
2218 value attribute "document-format-default" is set to 'application/octet-stream', the Printer object not only  
2219 supports auto-sensing of the document format, but will depend on the result of applying its auto-sensing  
2220 when the client does not supply the "document-format" attribute. If the client supplies a document  
2221 format value, the Printer **SHALLMUST** rely on the supplied attribute, rather than trust its auto-sensing  
2222 algorithm. To summarize:

- 2223 1. If the client does not supply a document format value, the Printer **MUST** rely on its default value  
2224 setting (which may be 'application/octet-stream' indicating an auto-sensing mechanism).
- 2225 2. If the client supplies a value other than 'application/octet-stream', the client is supplying valid  
2226 information about the format of the document data and the Printer object **SHALLMUST** trust the  
2227 client supplied value more than the outcome of applying an automatic format detection  
2228 mechanism. For example, the client may be requesting the printing of a PostScript file as a  
2229 'text/plain' document. The Printer object **SHALLMUST** print a text representation of the  
2230 PostScript commands rather than interpret the stream of PostScript commands and print the  
2231 result.
- 2232 3. If the client supplies a value of 'application/octet-stream', the client is indicating that the Printer  
2233 object **SHALLMUST** use its auto-sensing mechanism on the client supplied document data  
2234 whether auto-sensing is the Printer object's default or not.

2235  
2236 Note: Since the auto-sensing algorithm is probabilistic, if the client requests both auto-sensing  
2237 ("document-format" set to 'application/octet-stream') and true fidelity ("ipp-attribute-fidelity" set to  
2238 'true'), the Printer object might not be able to guarantee exactly what the end user intended (the auto-  
2239 sensing algorithm might mistake one document format for another ), but it is able to guarantee that its  
2240 auto-sensing mechanism be used.

2241 The maximum length of a 'mimeType' value in IPP is 255 octets.



## 2242 4.1.10 'octetString'

2243 The 'octetString' attribute syntax is a sequence of octets encoded in a maximum of 1023 octets which is  
2244 indicated in sub-section headers using the notation: octetString(MAX). This syntax type is used for  
2245 opaque data.

## 2246 4.1.11 'boolean'

2247 The 'boolean' attribute syntax is similar to an enum with only two values: 'true' and 'false'.

## 2248 4.1.12 'integer'

2249 The 'integer' attribute syntax is an integer value that is in the range from  $-2^{31}$  (MIN) to  $2^{31} - 1$   
2250 (MAX). Each individual attribute may specify the range constraint explicitly in sub-section headers if the  
2251 range is different from the full range of possible integer values. For example: job-priority  
2252 (integer(1:100)) for the "job-priority" attribute. However, the enforcement of that additional constraint is  
2253 up to the IPP objects, not the protocol.

## 2254 4.1.13 'rangeOfInteger'

2255 The 'rangeOfInteger' attribute syntax is an ordered pair of integers that defines an inclusive range of  
2256 integer values. The first integer specifies the lower bound and the second specifies the upper bound. If a  
2257 range constraint is specified in the header description for an attribute in this document whose attribute  
2258 syntax is 'rangeOfInteger' (i.e., 'X:Y' indicating X as a minimum value and Y as a maximum value), then  
2259 the constraint applies to both integers.

## 2260 4.1.14 'dateTime'

2261 The 'dateTime' attribute syntax is a standard, fixed length, 11 octet representation of the "DateAndTime"  
2262 syntax as defined in RFC 1903 [RFC1903]. RFC 1903 also identifies an 8 octet representation of a  
2263 "DateAndTime" value, but IPP objects MUST use the 11 octet representation. A user interface will  
2264 provide a mapping between protocol dateTime values and displayable user-friendly words or presentation  
2265 values and phrases which are localized to the natural language and date format of the user.

## 2266 4.1.15 'resolution'

2267 The 'resolution' attribute syntax specifies a two-dimensional resolution in the indicated units. It consists  
2268 of 3 integers: a cross feed direction resolution (positive integer value), a feed direction resolution  
2269 (positive integer value), and a units value. The semantics of these three components are taken from the  
2270 Printer MIB [RFC1759] suggested values. That is, the cross feed direction component resolution

2271 component is the same as the prtMarkerAddressabilityXFeedDir object in the Printer MIB, the feed  
2272 direction component resolution component is the same as the prtMarkerAddressabilityFeedDir in the  
2273 Printer MIB, and the units component is the same as the prtMarkerAddressabilityUnit object in the  
2274 Printer MIB (namely, '3' indicates dots per inch and '4' indicates dots per centimeter). All three values  
2275 MUST be present even if the first two values are the same. Example: '300', '600', '3' indicates a 300 dpi  
2276 cross-feed direction resolution, a 600 dpi feed direction resolution, since a '3' indicates dots per inch  
2277 (dpi).

#### 2278 4.1.16 '1setOf X'

2279 The '1setOf X' attribute syntax is 1 or more values of attribute syntax type X. This syntax type is used  
2280 for multi-valued attributes. The syntax type is called '1setOf' rather than just 'setOf' as a reminder that  
2281 the set of values **SHALLMUST** NOT be empty (i.e., a set of size 0). Sets are normally unordered.  
2282 However each attribute description of this type may specify that the values MUST be in a certain order  
2283 for that attribute.

## 2284 4.2 Job Template Attributes

2285 Job Template attributes describe job processing behavior. Support for Job Template attributes by a  
2286 Printer object is OPTIONAL (see section 13.2.3 for a description of support for OPTIONAL attributes).  
2287 Also, clients OPTIONALLY supply Job Template attributes in create requests.

2288 Job Template attributes conform to the following rules. For each Job Template attribute called "xxx":

- 2289 1. If the Printer object supports "xxx" then it **SHALLMUST** support both a "xxx-default" attribute  
2290 (unless there is a "No" in the table below) and a "xxx-supported" attribute. If the Printer object  
2291 doesn't support "xxx", then it **SHALLMUST** support neither an "xxx-default" attribute nor an  
2292 "xxx-supported" attribute, and it **SHALLMUST** treat an attribute "xxx" supplied by a client as  
2293 unsupported. An attribute "xxx" may be supported for some document formats and not  
2294 supported for other document formats. For example, it is expected that a Printer object would  
2295 only support "orientation-requested" for some document formats (such as 'text/plain' or  
2296 'text/html') but not others (such as 'application/postscript').  
2297
- 2298 2. "xxx" is OPTIONALLY supplied by the client in a create request. If "xxx" is supplied, the client is  
2299 indicating a desired job processing behavior for this Job. When "xxx" is not supplied, the client is  
2300 indicating that the Printer object apply its default job processing behavior at job processing time if  
2301 the document content does not contain an embedded instruction indicating an xxx-related  
2302 behavior.  
2303

2304 Note: Since an administrator MAY change the default value attribute after a Job object has been  
2305 submitted but before it has been processed, the default value used by the Printer object at job  
2306 processing time may be different that the default value in effect at job submission time.

2307

2308 3. The "xxx-supported" attribute is a Printer object attribute that describes which job processing  
2309 behaviors are supported by that Printer object. A client can query the Printer object to find out  
2310 what xxx-related behaviors are supported by inspecting the returned values of the "xxx-  
2311 supported" attribute.

2312

2313 Note: The "xxx" in each "xxx-supported" attribute name is singular, even though an "xxx-  
2314 supported" attribute usually has more than one value, such as "job-sheet-supported", unless the  
2315 "xxx" Job Template attribute is plural, such as "finishings" or "sides". In such cases the "xxx-  
2316 supported" attribute names are: "finishings-supported" and "sides-supported".

2317

2318 4. The "xxx-default" default value attribute describes what will be done at job processing time when  
2319 no other job processing information is supplied by the client (either explicitly as an IPP attribute in  
2320 the create request or implicitly as an embedded instruction within the document data).

2321

2322 If an application wishes to present an end user with a list of supported values from which to choose, the  
2323 application SHOULD query the Printer object for its supported value attributes. The application  
2324 SHOULD also query the default value attributes. If the application then limits selectable values to only  
2325 those value that are supported, the application can guarantee that the values supplied by the client in the  
2326 create request all fall within the set of supported values at the Printer. When querying the Printer, the  
2327 client MAY enumerate each attribute by name in the Get-Printer-Attributes Request, or the client MAY  
2328 just name the "job-template" group in order to get the complete set of supported attributes (both  
2329 supported and default attributes).

2330 The "finishings" attribute is an example of a Job Template attribute. It can take on a set of values such as  
2331 'staple', 'punch', and/or 'cover'. A client can query the Printer object for the "finishings-supported"  
2332 attribute and the "finishings-default" attribute. The supported attribute contains a set of supported  
2333 values. The default value attribute contains the finishing value(s) that will be used for a new Job if the  
2334 client does not supply a "finishings" attribute in the create request and the document data does not  
2335 contain any corresponding finishing instructions. If the client does supply the "finishings" attribute in the  
2336 create request, the IPP object validates the value or values to make sure that they are a subset of the  
2337 supported values identified in the Printer object's "finishings-supported" attribute. See section 3.2.1.2.

2338 The table below summarizes the names and relationships for all Job Template attributes. The first column  
2339 of the table (labeled "Job Attribute") shows the name and syntax for each Job Template attribute in the  
2340 Job object. These are the attributes that can optionally be supplied by the client in a create request. The  
2341 last two columns (labeled "Printer: Default Value Attribute" and "Printer: Supported Values Attribute")

2342 shows the name and syntax for each Job Template attribute in the Printer object (the default value  
2343 attribute and the supported values attribute). A "No" in the table means the Printer ~~SHALL~~MUST NOT  
2344 support the attribute (that is, the attribute is simply not applicable). For brevity in the table, the 'text' and  
2345 'name' entries do not show the maximum length for each attribute, as in "(127)".

2346	+	=====+	=====+	=====+
2347		Job Attribute	Printer: Default Value	Printer: Supported
2348			Attribute	Values Attribute
2349		=====+	=====+	=====+
2350		job-priority	job-priority-default	job-priority-supported
2351		(integer 1:100)	(integer 1:100)	(integer 1:100)
2352		-----+	-----+	-----+
2353		job-hold-until	job-hold-until-	job-hold-until-
2354		(type3 keyword	default	supported
2355		name)	(type3 keyword	(1setOf
2356			name)	type3 keyword   name)
2357		-----+	-----+	-----+
2358		job-sheets	job-sheets-default	job-sheets-supported
2359		(type3 keyword	(type3 keyword	(1setOf
2360		name)	name)	type3 keyword   name)
2361		-----+	-----+	-----+
2362		multiple-document-	multiple-document-	multiple-document-
2363		handling	handling-default	handling-supported
2364		(type2 keyword)	(type2 keyword)	(1setOf type2 keyword)
2365		-----+	-----+	-----+
2366		copies	copies-default	copies-supported
2367		(integer (1:MAX))	(integer (1:MAX))	(rangeOfInteger
2368				(1:MAX))
2369		-----+	-----+	-----+
2370		finishings	finishings-default	finishings-supported
2371		(1setOf type2 enum)	(1setOf type2 enum)	(1setOf type2 enum)
2372		-----+	-----+	-----+
2373		page-ranges	No	page-ranges-
2374		(1setOf		supported (boolean)
2375		rangeOfInteger		
2376		(1:MAX))		
2377		-----+	-----+	-----+
2378		sides	sides-default	sides-supported
2379		(type2 keyword)	(type2 keyword)	(1setOf type2 keyword)
2380		-----+	-----+	-----+
2381		number-up	number-up-default	number-up-supported
2382		(integer (1:MAX))	(integer (1:MAX))	(1setOf integer
2383				(1:MAX)
2384				rangeOfInteger
2385				(1:MAX))
2386		-----+	-----+	-----+
2387		orientation-	orientation-requested-	orientation-requested-
2388		requested	default	supported
2389		(type2 enum)	(type2 enum)	(1setOf type2 enum)
2390		-----+	-----+	-----+

2391	media	media-default	media-supported
2392	(type3 keyword	(type3 keyword	(1setOf
2393	name)	name)	type3 keyword   name)
2394			
2395			media-ready
2396			(1setOf
2397			type3 keyword   name)
2398	+-----+-----+-----+		
2399	printer-resolution	printer-resolution-	printer-resolution-
2400	(resolution)	default	supported
2401		(resolution)	(1setOf resolution)
2402	+-----+-----+-----+		
2403	print-quality	print-quality-default	print-quality-
2404	(type2 enum)	(type2 enum)	supported
2405			(1setOf type2 enum)
2406	+-----+-----+-----+		
2407			
2408			

#### 2409 4.2.1 job-priority (integer(1:100))

2410 This attribute specifies a priority for scheduling the Job. A higher value specifies a higher priority. The  
 2411 value 1 indicates the lowest possible priority. The value 100 indicates the highest possible priority.  
 2412 Among those jobs that are ready to print, a Printer **SHALLMUST** print all jobs with a priority value of n  
 2413 before printing those with a priority value of n-1 for all n.

2414 If the Printer object supports this attribute, it **SHALLMUST** always support the full range from 1 to 100.  
 2415 No administrative restrictions are permitted. This way an end-user can always make full use of the entire  
 2416 range with any Printer object. If privileged jobs are implemented outside IPP/1.0, they **SHALLMUST**  
 2417 have priorities higher than 100, rather than restricting the range available to end-users.

2418 If the client does not supply this attribute and this attribute is supported by the Printer object, the Printer  
 2419 object **SHALLMUST** use the value of the Printer object's "job-priority-default" at job submission time  
 2420 (unlike most Job Template attributes that are used if necessary at job processing time).

2421 The syntax for the "job-priority-supported" is also integer(1:100). This single integer value indicates the  
 2422 number of priority levels supported. The Printer object **SHALLMUST** take the value supplied by the  
 2423 client and map it to the closest integer in a sequence of n integers values that are evenly distributed over  
 2424 the range from 1 to 100 using the formula:

2425 
$$\text{roundToNearestInt}((100x+50)/n)$$

2426 where n is the value of "job-priority-supported" and x ranges from 0 through n-1.

2427 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,  
2428 the sequence of values is: 17, 50 and 83; if n = 10, the sequence of values is: 5, 15, 25, 35, 45, 55, 65,  
2429 75, 85, and 95; if n = 100, the sequence of values is: 1, 2, 3, ... 100.

2430 If the value of the Printer object's "job-priority-supported" is 10 and the client supplies values in the range  
2431 1 to 10, the Printer object maps them to 5, in the range 11 to 20, the Printer object maps them to 15, etc.

#### 2432 4.2.2 job-hold-until (type3 keyword | name (MAX))

2433 This attribute specifies the named time period during which the Job **SHALLMUST** become a candidate  
2434 for printing.

2435 Standard values for named time periods are:

2436 'no-hold': immediately, if there are not other reasons to hold the job

2437 'day-time': during the day

2438 'evening': evening

2439 'night': night

2440 'weekend': weekend

2441 'second-shift': second-shift (after close of business)

2442 'third-shift': third-shift (after midnight)

2443

2444 An administrator **SHALLMUST** associate allowable print times with a named time period (by means  
2445 outside IPP/1.0). An administrator is encouraged to pick names that suggest the type of time period. An  
2446 administrator **MAY** define additional values using the 'name' or 'keyword' attribute syntax, depending on  
2447 implementation.

2448 If the value of this attribute specifies a time period that is in the future, the Printer **SHALLMUST** add the  
2449 'job-hold-until-specified' value to the job's "job-state-reasons" attribute, move the job to the 'pending-held'  
2450 state, and **SHALLMUST** NOT schedule the job for printing until the specified time-period arrives. When  
2451 the specified time period arrives, the Printer **SHALLMUST** remove the 'job-hold-until-specified' value  
2452 from the job's "job-state-reason" attribute and, if there are no other job state reasons that keep the job in  
2453 the 'pending-held' state, the Printer **SHALLMUST** consider the job as a candidate for processing by  
2454 moving the job to the 'pending' state.

2455 If this job attribute value is the named value 'no-hold', or the specified time period has already started, the  
2456 job **SHALLMUST** be a candidate for processing immediately.

2457 If the client does not supply this attribute and this attribute is supported by the Printer object, the Printer  
2458 object **SHALLMUST** use the value of the Printer object's "job-hold-until-default" at job submission time  
2459 (unlike most Job Template attributes that are used if necessary at job processing time).

## 2460 4.2.3 job-sheets (type3 keyword | name(MAX))

2461 This attribute determines which job start/end sheet(s), if any, **SHALLMUST** be printed with a job.

2462 Standard values are:

2463 'none': no job sheet is printed

2464 'standard': one or more site specific standard job sheets are printed, e.g. a single start sheet or both  
2465 start and end sheet is printed

2466  
2467 An administrator MAY define additional values using the 'name' or 'keyword' attribute syntax, depending  
2468 on implementation.

2469 Note: The effect of this attribute on jobs with multiple documents MAY be affected by the "multiple-  
2470 document-handling" job attribute (section 4.2.4), depending on the job sheet semantics.

## 2471 4.2.4 multiple-document-handling (type2 keyword)

2472 This attribute is relevant only if a job consists of two or more documents. The attribute controls finishing  
2473 operations and the placement of one or more print-stream pages into impressions and onto media sheets.  
2474 When the value of the "copies" attribute exceeds 1, it also controls the order in which the copies that  
2475 result from processing the documents are produced. For the purposes of this explanations, if "a"  
2476 represents an instance of document data, then the result of processing the data in document "a" is a  
2477 sequence of media sheets represented by "a(\*)".

2478 Standard values are:

2479 'single-document': If a Job object has multiple documents, say, the document data is called a and b,  
2480 then the result of processing all the document data (a and then b) **SHALLMUST** be treated as a  
2481 single sequence of media sheets for finishing operations; that is, finishing would be performed on  
2482 the concatenation of the sequences a(\*),b(\*). The Printer object **SHALLMUST** NOT force the  
2483 data in each document instance to be formatted onto a new print-stream page, nor to start a new  
2484 impression on a new media sheet. If more than one copy is made, the ordering of the sets of media  
2485 sheets resulting from processing the document data **SHALLMUST** be a(\*), b(\*), a(\*), b(\*), ...,  
2486 and the Printer object **SHALLMUST** force each copy (a(\*),b(\*)) to start on a new media sheet.  
2487 'separate-documents-uncollated-copies': If a Job object has multiple documents, say, the document  
2488 data is called a and b, then the result of processing the data in each document instance  
2489 **SHALLMUST** be treated as a single sequence of media sheets for finishing operations; that is, the  
2490 sets a(\*) and b(\*) would each be finished separately. The Printer object **SHALLMUST** force each  
2491 copy of the result of processing the data in a single document to start on a new media sheet. If



2492 more than one copy is made, the ordering of the sets of media sheets resulting from processing the  
2493 document data **SHALLMUST** be a(\*), a(\*), ..., b(\*), b(\*) ... .

2494 'separate-documents-collated-copies': If a Job object has multiple documents, say, the document data  
2495 is called a and b, then the result of processing the data in each document instance **SHALLMUST**  
2496 be treated as a single sequence of media sheets for finishing operations; that is, the sets a(\*) and  
2497 b(\*) would each be finished separately. The Printer object **SHALLMUST** force each copy of the  
2498 result of processing the data in a single document to start on a new media sheet. If more than one  
2499 copy is made, the ordering of the sets of media sheets resulting from processing the document  
2500 data **SHALLMUST** be a(\*), b(\*), a(\*), b(\*), ... .

2501

2502 The 'single-document' value is the same as 'separate-documents-collated-copies' with respect to ordering  
2503 of print-stream pages, but not media sheet generation, since 'single-document' will put the first page of  
2504 the next document on the back side of a sheet if an odd number of pages have been produced so far for  
2505 the job, while 'separate-documents-collated-copies' always forces the next document or document copy  
2506 on to a new sheet. In addition, if the "finishings" attribute specifies 'staple', then with 'single-document',  
2507 documents a and b are stapled together as a single document, but with 'separate-documents-uncollated-  
2508 copies' and 'separate-documents-collated-copies', documents a and b are stapled separately.

2509 Note: None of these values provide means to produce uncollated sheets within a document, i.e., where  
2510 multiple copies of sheet n are produced before sheet n+1 of the same document.

2511 The relationship of this attribute and the other attributes that control document processing is described in  
2512 section 16.5.

#### 2513 4.2.5 copies (integer(1:MAX))

2514 This attribute specifies the number of copies to be printed.

2515 On many devices the supported number of collated copies will be limited by the number of physical  
2516 output bins on the device, and may be different from the number of uncollated copies which can be  
2517 supported.

2518 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-  
2519 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other  
2520 attributes that control document processing is described in section 16.5.

#### 2521 4.2.6 finishings (1setOf type2 enum)

2522 This attribute identifies the finishing operations that the Printer uses for each copy of each printed  
2523 document in the Job. For Jobs with multiple documents, the "multiple-document-handling" attribute  
2524 determines what constitutes a "copy" for purposes of finishing.

2525 Standard values are:

2526	Value	Symbolic Name and Description
2527		
2528	'3'	'none': Perform no finishing
2529	'4'	'staple': Bind the document(s) with one or more staples. The exact number and placement
2530		of the staples is site-defined.
2531	'5'	'punch': This value indicates that holes are required in the finished document. The exact
2532		number and placement of the holes is site-defined. The punch specification MAY
2533		be satisfied (in a site- and implementation-specific manner) either by
2534		drilling/punching, or by substituting pre-drilled media.
2535	'6'	'cover': This value is specified when it is desired to select a non-printed (or pre-printed)
2536		cover for the document. This does not supplant the specification of a printed cover
2537		(on cover stock medium) by the document itself.
2538	'7'	'bind': This value indicates that a binding is to be applied to the document; the type and
2539		placement of the binding is site-defined."

2541 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-  
2542 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other  
2543 attributes that control document processing is described in section 16.5.

2544 If the client supplies a value of 'none' along with any other combination of values, it is the same as if only  
2545 that other combination of values had been supplied (that is the 'none' value has no effect).

#### 2546 4.2.7 page-ranges (1setOf rangeOfInteger (1:MAX))

2547 This attribute identifies the range(s) of print-stream pages that the Printer object uses for each copy of  
2548 each document which are to be printed. Nothing is printed for any pages identified that do not exist in  
2549 the document(s). Ranges **SHALLMUST** be in ascending order, for example: 1-3, 5-7, 15-19 and  
2550 **SHALLMUST** NOT overlap, so that a non-spooling Printer object can process the job in a single pass. If  
2551 the ranges are not ascending or are overlapping, the IPP object **SHALLMUST** reject the request and  
2552 return the 'client-error-bad-request' status code. The attribute is associated with print-stream pages not  
2553 application-numbered pages (for example, the page numbers found in the headers and or footers for  
2554 certain word processing applications).

2555 For Jobs with multiple documents, the "multiple-document-handling" attribute determines what  
2556 constitutes a "copy" for purposes of the specified page range(s). When "multiple-document-handling" is  
2557 'single-document', the Printer object **SHALLMUST** apply each supplied page range once to the  
2558 concatenation of the print-stream pages. For example, if there are 8 documents of 10 pages each, the  
2559 page-range '41:60' prints the pages in the 5th and 6th documents as a single document and none of the  
2560 pages of the other documents are printed. When "multiple-document-handling" is 'separate-document-

2561 uncollated-copies' or 'separate-document-collated-copies', the Printer object ~~SHALL~~**MUST** apply each  
2562 supplied page range repeatedly to each document copy. For the same job, the page-range '1:3, 10:10'  
2563 would print the first 3 pages and the 10th page of each of the 8 documents in the Job, as 8 separate  
2564 documents.

2565 In most cases, the exact pages to be printed will be generated by a device driver and this attribute would  
2566 not be required. However, when printing an archived document which has already been formatted, the  
2567 end user may elect to print just a subset of the pages contained in the document. In this case, if page-  
2568 range = n.m is specified, the first page to be printed will be page n. All subsequent pages of the document  
2569 will be printed through and including page m.

2570 "page-ranges-supported" is a boolean value indicating whether or not the printer is capable of supporting  
2571 the printing of page ranges. This capability may differ from one PDL to another. There is no "page-  
2572 ranges-default" attribute. If the "page-ranges" attribute is not supplied by the client, all pages of the  
2573 document will be printed.

2574 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-  
2575 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other  
2576 attributes that control document processing is described in section 16.5.

#### 2577 4.2.8 sides (type2 keyword)

2578 This attribute specifies how print-stream pages are to be imposed upon the sides of an instance of a  
2579 selected medium, i.e., an impression.

2580 The standard values are:

2581 'one-sided': imposes each consecutive print-stream page upon the same side of consecutive media  
2582 sheets.

2583 'two-sided-long-edge': imposes each consecutive pair of print-stream pages upon front and back sides  
2584 of consecutive media sheets, such that the orientation of each pair of print-stream pages on the  
2585 medium would be correct for the reader as if for binding on the long edge. This imposition is  
2586 sometimes called 'duplex' or 'head-to-head'.

2587 'two-sided-short-edge': imposes each consecutive pair of print-stream pages upon front and back sides  
2588 of consecutive media sheets, such that the orientation of each pair of print-stream pages on the  
2589 medium would be correct for the reader as if for binding on the short edge. This imposition is  
2590 sometimes called 'tumble' or 'head-to-toe'.

2591  
2592 'two-sided-long-edge', 'two-sided-short-edge', 'tumble', and 'duplex' all work the same for portrait or  
2593 landscape. However 'head-to-toe' is 'tumble' in portrait but 'duplex' in landscape. 'head-to-head' also  
2594 switches between 'duplex' and 'tumble' when using portrait and landscape modes.

2595 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-  
 2596 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other  
 2597 attributes that control document processing is described in section 16.5.

#### 2598 4.2.9 number-up (integer(1:MAX))

2599 This attribute specifies the number of print-stream pages to impose upon a single side of an instance of a  
 2600 selected medium. For example, if the value is

2601	Value	Description
2602		
2603	'1'	The Printer <b>SHALLMUST</b> place one print-stream page on a single side of an instance of
2604		the selected medium (MAY add some sort of translation, scaling, or rotation).
2605	'2'	The Printer <b>SHALLMUST</b> place two print-stream pages on a single side of an instance of
2606		the selected medium (MAY add some sort of translation, scaling, or rotation).
2607	'4'	The Printer <b>SHALLMUST</b> place four print-stream pages on a single side of an instance of
2608		the selected medium (MAY add some sort of translation, scaling, or rotation).
2609		

2610 This attribute primarily controls the translation, scaling and rotation of print-stream pages.

2611 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-  
 2612 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other  
 2613 attributes that control document processing is described in section 16.5.

#### 2614 4.2.10 orientation-requested (type2 enum)

2615 This attribute indicates the desired orientation for printed print-stream pages; it does not describe the  
 2616 orientation of the client-supplied print-stream pages.

2617 For some document formats (such as 'application/postscript'), the desired orientation of the print-stream  
 2618 pages is specified within the document data. This information is generated by a device driver prior to the  
 2619 submission of the print job. Other document formats (such as 'text/plain') do not include the notion of  
 2620 desired orientation within the document data. In the latter case it is possible for the Printer object to bind  
 2621 the desired orientation to the document data after it has been submitted. It is expected that a Printer  
 2622 object would only support "orientations-requested" for some document formats (e.g., 'text/plain' or  
 2623 'text/html') but not others (e.g., 'application/postscript'). This is no different than any other Job Template  
 2624 attribute since section 4.2, item 1, points out that a Printer object may support or not support any Job  
 2625 Template attribute based on the document format supplied by the client. However, a special mention is  
 2626 made here since it is very likely that a Printer object will support "orientation-requested" for only a subset  
 2627 of the supported document formats.

2628 Standard values are:

2629	Value	Symbolic Name and Description
2630		
2631	'3'	'portrait': The content will be imaged across the short edge of the medium.
2632	'4'	'landscape': The content will be imaged across the long edge of the medium. Landscape is
2633		defined to be a rotation of the print-stream page to be imaged by +90 degrees with
2634		respect to the medium (i.e. anti-clockwise) from the portrait orientation. Note:
2635		The +90 direction was chosen because simple finishing on the long edge is the
2636		same edge whether portrait or landscape
2637	'5'	'reverse-landscape': The content will be imaged across the long edge of the medium.
2638		Reverse-landscape is defined to be a rotation of the print-stream page to be imaged
2639		by -90 degrees with respect to the medium (i.e. clockwise) from the portrait
2640		orientation. Note: The 'reverse-landscape' value was added because some
2641		applications rotate landscape -90 degrees from portrait, rather than +90 degrees.
2642	'6'	'reverse-portrait': The content will be imaged across the shshort edge of the medium.
2643		Reverse-portrait is defined to be a rotation of the print-stream page to be imaged
2644		by 180 degrees with respect to the medium from the portrait orientation. Note:
2645		The 'reverse-portrait' value was added for use with the "finishings" attribute in
2646		cases where the opposite edge is desired for finishing a portrait document on
2647		simple finishing devices that have only one finishing position. Thus a 'text/plain'
2648		portrait document can be stapled "on the right" by a simple finishing device as is
2649		common use with some middle eastern languages such as Hebrew.

2651 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-  
 2652 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other  
 2653 attributes that control document processing is described in section 16.5.

#### 2654 4.2.11 media (type3 keyword | name(MAX))

2655 This attribute identifies the medium that the Printer uses for all impressions of the Job.

2656 The values for "media" include medium-names, medium-sizes, input-trays and electronic forms so that  
 2657 one attribute specifies the media. If a Printer object supports a medium name as a value of this attribute,  
 2658 such a medium name implicitly selects an input-tray that contains the specified medium. If a Printer  
 2659 object supports a medium size as a value of this attribute, such a medium size implicitly selects a medium  
 2660 name that in turn implicitly selects an input-tray that contains the medium with the specified size. If a  
 2661 Printer object supports an input-tray as the value of this attribute, such an input-tray implicitly selects the  
 2662 medium that is in that input-tray at the time the job prints. This case includes manual-feed input-trays. If  
 2663 a Printer object supports an electronic form as the value of this attribute, such an electronic form

2664 implicitly selects a medium-name that in turn implicitly selects an input-tray that contains the medium  
 2665 specified by the electronic form. The electronic form also implicitly selects an image that the Printer  
 2666 ~~SHALL~~**MUST** merge with the document data as its prints each page.

2667 Standard values are (taken from ISO DPA and the Printer MIB) and are listed in section 15. An  
 2668 administrator **MAY** define additional values using the 'name' or 'keyword' attribute syntax, depending on  
 2669 implementation.

2670 There is also an additional Printer attribute named "media-ready" which differs from "media-supported" in  
 2671 that legal values only include the subset of "media-supported" values that are physically loaded and ready  
 2672 for printing with no operator intervention required. If an IPP object supports "media-supported", it  
 2673 **NEED NOT** support "media-ready".

2674 The relationship of this attribute and the other attributes that control document processing is described in  
 2675 section 16.5.

#### 2676 4.2.12 printer-resolution (resolution)

2677 This attribute identifies the resolution that Printer uses for the Job.

#### 2678 4.2.13 print-quality (type2 enum)

2679 This attribute specifies the print quality that the Printer uses for the Job.

2680 The standard values are:

2681	Value	Symbolic Name and Description
2682		
2683	'3'	'draft': lowest quality available on the printer
2684	'4'	'normal': normal or intermediate quality on the printer
2685	'5'	'high': highest quality available on the printer
2686		

### 2687 4.3 Job Description Attributes

2688 The attributes in this section form the attribute group called "job-description". The following table  
 2689 summarizes these attributes. The third column indicates whether the attribute is a  
 2690 ~~MANDATORYREQUIRED~~ attribute that **MUST** be supported by Printer objects. If it is not indicated  
 2691 as ~~MANDATORYREQUIRED~~, then it is **OPTIONAL**. The maximum size in octets for 'text' and 'name'  
 2692 attributes is indicated in parentheses.

	Attribute	Syntax
2693		
2694		
2695	<u>MANDATORYREQUIRED?</u>	
2696		
2697	job-uri	uri
2698	<u>MANDATORYREQUIRED</u>	
2699		
2700	job-id	integer(1:MAX)
2701	<u>MANDATORYREQUIRED</u>	
2702		
2703	job-printer-uri	uri
2704	<u>MANDATORYREQUIRED</u>	
2705		
2706	job-more-info	uri
2707		
2708	job-name	name (MAX)
2709	<u>MANDATORYREQUIRED</u>	
2710		
2711	job-originating-user-name	name (MAX)
2712	<u>MANDATORYREQUIRED</u>	
2713		
2714	job-state	type1 enum
2715	<u>MANDATORYREQUIRED</u>	
2716		
2717	job-state-reasons	1setOf type2 keyword
2718		
2719	job-state-message	text (MAX)
2720		
2721	number-of-documents	integer (0:MAX)
2722		
2723	output-device-assigned	name (127)
2724		
2725	time-at-creation	integer (0:MAX)
2726		
2727	time-at-processing	integer (0:MAX)
2728		
2729	time-at-completed	integer (0:MAX)
2730		
2731	number-of-intervening-jobs	integer (0:MAX)
2732		
2733	job-message-from-operator	text (127)
2734		
2735	job-k-octets	integer (0:MAX)
2736		
2737	job-impressions	integer (0:MAX)

2738	+-----+-----+-----+
2739	job-media-sheets   integer (0:MAX)
2740	+-----+-----+-----+
2741	job-k-octets-processed   integer (0:MAX)
2742	+-----+-----+-----+
2743	job-impressions-completed   integer (0:MAX)
2744	+-----+-----+-----+
2745	job-media-sheets-completed   integer (0:MAX)
2746	+-----+-----+-----+
2747	attributes-charset   charset
2748	<del>MANDATORY</del> <u>REQUIRED</u>
2749	+-----+-----+-----+
2750	attributes-natural-language   naturalLanguage   <u>REQUIRED</u>
2751	+-----+-----+-----+
2752	
2753	

#### 2754 4.3.1 job-uri (uri)

2755 This **REQUIRED** attribute contains the URI for the job. The Printer object, on receipt of a new job,  
 2756 generates a URI which identifies the new Job. The Printer object returns the value of the "job-uri"  
 2757 attribute as part of the response to a create request. The precise format of a Job URI is implementation  
 2758 dependent. If the Printer object supports more than one URI and there is some relationship between the  
 2759 newly formed Job URI and the Printer object's URI, the Printer object uses the Printer URI supplied by  
 2760 the client in the create request. For example, if the create request comes in over a secure channel, the  
 2761 new Job URI **MUST** use the same secure channel. This can **be** guaranteed because the Printer object is  
 2762 responsible for generating the Job URI **and the Printer object** is aware of its security configuration and  
 2763 policy as well as the Printer URI used in the create request.

2764 For a description of this attribute and its relationship to "job-id" and "job-printer-uri" attribute, see the  
 2765 discussion in section 2.4 on "Object Identity".

#### 2766 4.3.2 job-id (integer(1:MAX))

2767 This **REQUIRED** attribute contains the ID of the job. The Printer, on receipt of a new job, generates an  
 2768 ID which identifies the new Job on that Printer. The Printer returns the value of the "job-id" attribute as  
 2769 part of the response to a create request. The 0 value is not included to allow for compatibility with  
 2770 SNMP index values which also cannot be 0.

2771 For a description of this attribute and its relationship to "job-uri" and "job-printer-uri" attribute, see the  
 2772 discussion in section 2.4 on "Object Identity".



## 2773 4.3.3 job-printer-uri (uri)

2774 This REQUIRED attribute identifies the Printer object that created this Job object. When a Printer object  
2775 creates a Job object, it populates this attribute with the Printer object URI that was used in the create  
2776 request. This attribute permits a client to identify the Printer object that created this Job object when only  
2777 the Job object's URI is available to the client. The client queries the creating Printer object to determine  
2778 which languages, charsets, operations, are supported for this Job.

2779 For a description of this attribute and its relationship to "job-uri" and "job-id" attribute, see the discussion  
2780 in section 2.4 on "Object Identity".

## 2781 4.3.4 job-more-info (uri)

2782 Similar to "printer-more-info", this attribute contains the URI referencing some resource with more  
2783 information about this Job object, perhaps an HTML page containing information about the Job.

## 2784 4.3.5 job-name (name(MAX))

2785 This REQUIRED attribute is the name of the job. It is a name that is more user friendly than the "job-  
2786 uri" attribute value. It does not need to be unique between Jobs. The Job's "job-name" attribute is set to  
2787 the value supplied by the client in the "job-name" operation attribute in the create request (see Section  
2788 3.2.1.1). If, however, the "job-name" operation attribute is not supplied by the client in the create  
2789 request, the Printer object, on creation of the Job, ~~SHALL~~**MUST** generate a name. The printer  
2790 SHOULD generate the value of the Job's "job-name" attribute from the first of the following sources that  
2791 produces a value: 1) the "document-name" operation attribute of the first (or only) document, 2) the  
2792 "document-URI" attribute of the first (or only) document, or 3) any other piece of Job specific and/or  
2793 Document Content information.

## 2794 4.3.6 job-originating-user-name (name(MAX))

2795 This REQUIRED attribute contains the name of the end user that submitted the print job. The Printer  
2796 object sets this attribute to the most authenticated printable name that it can obtain from the  
2797 authentication service over which the IPP operation was received. Only if such is not available, does the  
2798 Printer object use the value supplied by the client in the "requesting-user-name" operation attribute of the  
2799 create operation (see Section 8).

2800 Note: The Printer object needs to keep an internal originating user id of some form, typically as a  
2801 credential of a principal, with the Job object. Since such an internal attribute is implementation-  
2802 dependent and not of interest to clients, it is not specified as a Job Description attribute. This originating  
2803 user id is used for authorization checks (if any) on all subsequent operation.

## 2804 4.3.7 job-state (type1 enum)

2805 This REQUIRED attribute identifies the current state of the job. Even though the IPP protocol defines  
 2806 eight values for job states, implementations only need to support those states which are appropriate for  
 2807 the particular implementation. In other words, a Printer supports only those job states implemented by  
 2808 the output device and available to the Printer object implementation.

2809 Standard values are:

2810 Values      Symbolic Name and Description

2811

2812 '3'          'pending': The job is a candidate to start processing, but is not yet processing.

2813

2814 '4'          'pending-held': The job is not a candidate for processing for any number of reasons but  
 2815 will return to the 'pending' state as soon as the reasons are no longer present. The  
 2816 job's "job-state-reason" attribute **SHALLMUST** indicate why the job is no longer a  
 2817 candidate for processing.

2818

2819 '5'          'processing': One or more of:

2820

2821 1. the job is using, or is attempting to use, one or more purely software processes  
 2822 that are analyzing, creating, or interpreting a PDL, etc.,

2823 2. the job is using, or is attempting to use, one or more hardware devices that are  
 2824 interpreting a PDL, making marks on a medium, and/or performing finishing, such  
 2825 as stapling, etc.,

2826 3. the Printer object has made the job ready for printing, but the output device is  
 2827 not yet printing it, either because the job hasn't reached the output device or  
 2828 because the job is queued in the output device or some other spooler, awaiting the  
 2829 output device to print it.

2830

2831 When the job is in the 'processing' state, the entire job state includes the detailed  
 2832 status represented in the printer's "printer-state", "printer-state-reasons", and  
 2833 "printer-state-message" attributes.

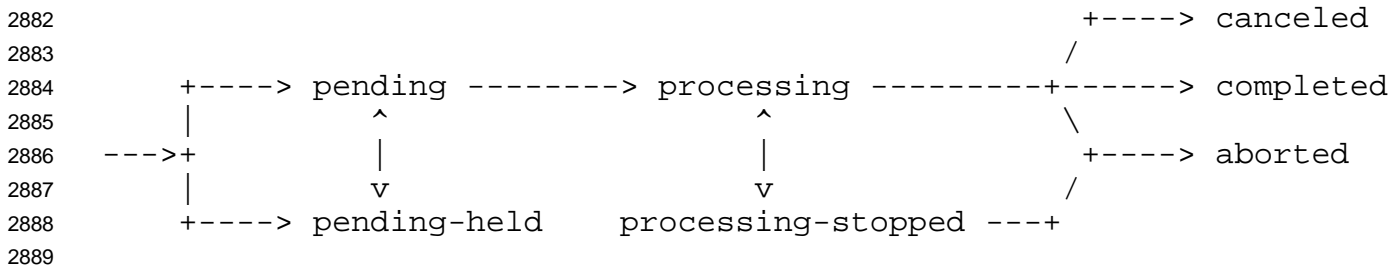
2834 Implementations MAY, though they NEED NOT, include additional values in the  
 2835 job's "job-state-reasons" attribute to indicate the progress of the job, such as  
 2836 adding the 'job-printing' value to indicate when the output device is actually  
 2837 making marks on paper and/or the 'processing-to-stop-point' value to indicate that  
 2838 the IPP object is in the process of canceling or aborting the job. Most  
 2839 implementations won't bother with this nuance.

2840

- 2841 '6' 'processing-stopped': The job has stopped while processing for any number of reasons and  
2842 will return to the 'processing' state as soon as the reasons are no longer present.  
2843  
2844 The job's "job-state-reason" attribute MAY indicate why the job has stopped  
2845 processing. For example, if the output device is stopped, the 'printer-stopped'  
2846 value MAY be included in the job's "job-state-reasons" attribute.  
2847  
2848 Note: When an output device is stopped, the device usually indicates its condition  
2849 in human readable form locally at the device. A client can obtain more complete  
2850 device status remotely by querying the Printer object's "printer-state", "printer-  
2851 state-reasons" and "printer-state-message" attributes.  
2852  
2853 '7' 'canceled': The job has been canceled by a Cancel-Job operation and the Printer object has  
2854 completed canceling the job and all job status attributes have reached their final  
2855 values for the job. While the Printer object is canceling the job, the job remains in  
2856 its current state, but the job's "job-state-reasons" attribute SHOULD contain the  
2857 'processing-to-stop-point' value and one of the 'canceled-by-user', 'canceled-by-  
2858 operator', or 'canceled-at-device' value. When the job moves to the 'canceled'  
2859 state, the 'processing-to-stop-point' value, if present, **SHALLMUST** be removed,  
2860 but the 'canceled-by-xxx', if present, **SHALLMUST** remain.  
2861  
2862 '8' 'aborted': The job has been aborted by the system, usually while the job was in the  
2863 'processing' or 'processing-stopped' state and the Printer has completed aborting  
2864 the job and all job status attributes have reached their final values for the job.  
2865 While the Printer object is aborting the job, the job remains in its current state, but  
2866 the job's "job-state-reasons" attribute SHOULD contain the 'processing-to-stop-  
2867 point' and 'aborted-by-system' values. When the job moves to the 'aborted' state,  
2868 the 'processing-to-stop-point' value, if present, **SHALLMUST** be removed, but  
2869 the 'aborted-by-system' value, if present, **SHALLMUST** remain.  
2870  
2871 '9' 'completed': The job has completed successfully or with warnings or errors after  
2872 processing and all of the job media sheets have been successfully stacked in the  
2873 appropriate output bin(s) and all job status attributes have reached their final  
2874 values for the job. The job's "job-state-reasons" attribute SHOULD contain one  
2875 of: 'completed-successfully', 'completed-with-warnings', or 'completed-with-errors'  
2876 values.  
2877

2878 The final value for this attribute **SHALLMUST** be one of: 'completed', 'canceled', or 'aborted' before the  
 2879 Printer removes the job altogether. The length of time that jobs remain in the 'canceled', 'aborted', and  
 2880 'completed' states depends on implementation.

2881 The following figure shows the normal job state transitions.



2890 Normally a job progresses from left to right. Other state transitions are unlikely, but are not forbidden.  
 2891 Not shown are the transitions to the 'canceled' state from the 'pending', 'pending-held', and 'processing-  
 2892 stopped' states.

2893 Jobs reach one of the three terminal states: 'completed', 'canceled', or 'aborted', after the jobs have  
 2894 completed all activity, including stacking output media, after the jobs have completed all activity, and all  
 2895 job status attributes have reached their final values for the job.

2896 Note: As with all other IPP attributes, if the implementation can not determine the correct value for this  
 2897 attribute, it ~~may choose to~~ **SHOULD** respond with the out-of-band value 'unknown' (see section 4.1)  
 2898 rather than try to guess at some possibly incorrect value and give the end user the wrong impression  
 2899 about the state of the Job object. For example, if the implementation is just a gateway into some printing  
 2900 system that does not provide detailed status about the print job, the IPP Job object's state might literally  
 2901 be 'unknown'.

#### 2902 4.3.8 job-state-reasons (1setOf type2 keyword)

2903 This attribute provides additional information about the job's current state, i.e., information that augments  
 2904 the value of the job's "job-state" attribute.

2905 Implementation of these values is **OPTIONAL**, i.e., a Printer **NEED NOT** implement them, even if (1)  
 2906 the output device supports the functionality represented by the reason and (2) is available to the Printer  
 2907 object implementation. These values **MAY** be used with any job state or states for which the reason  
 2908 makes sense. Furthermore, when implemented, the Printer **SHALLMUST** return these values when the  
 2909 reason applies and **SHALLMUST NOT** return them when the reason no longer applies whether the value  
 2910 of the Job's "job-state" attribute changed or not. When the Job does not have any reasons for being in its  
 2911 current state, the value of the Job's "job-state-reasons" attribute **SHALLMUST** be 'none'.

2912 Note: While values cannot be added to the 'job-state' attribute without impacting deployed clients that  
2913 take actions upon receiving "job-state" values, it is the intent that additional "job-state-reasons" values  
2914 can be defined and registered without impacting such deployed clients. In other words, the "job-state-  
2915 reasons" attribute is intended to be extensible.

2916 The following standard values are defined. For ease of understanding, the values are presented in the  
2917 order in which the reasons are likely to occur (if implemented), starting with the 'job-incoming' value:

2918 'none': There are no reasons for the job's current state.

2919 'job-incoming': The Create-Job operation has been accepted by the Printer, but the Printer is  
2920 expecting additional Send-Document and/or Send-URI operations and/or is accessing/accepting  
2921 document data.

2922 'submission-interrupted': The job was not completely submitted for some unforeseen reason, such as:  
2923 (1) the Printer has crashed before the job was closed by the client, (2) the Printer or the document  
2924 transfer method has crashed in some non-recoverable way before the document data was entirely  
2925 transferred to the Printer, (3) the client crashed or failed to close the job before the time-out  
2926 period.

2927 'job-outgoing': The Printer is transmitting the job to the output device.

2928 'job-hold-until-specified': The value of the job's "job-hold-until" attribute was specified with a time  
2929 period that is still in the future. The job ~~SHALL~~MUST NOT be a candidate for processing until  
2930 this reason is removed and there are no other reasons to hold the job.

2931 'resources-are-not-ready': At least one of the resources needed by the job, such as media, fonts,  
2932 resource objects, etc., is not ready on any of the physical printer's for which the job is a candidate.  
2933 This condition MAY be detected when the job is accepted, or subsequently while the job is  
2934 pending or processing, depending on implementation. The job may remain in its current state or  
2935 be moved to the 'pending-held' state, depending on implementation and/or job scheduling policy.

2936 'printer-stopped-partly': The value of the Printer's "printer-state-reasons" attribute contains the value  
2937 'stopped-partly'.

2938 'printer-stopped': The value of the Printer's "printer-state" attribute is 'stopped'.

2939 'job-interpreting': Job is in the 'processing' state, but more specifically, the Printer is interpreting the  
2940 document data.

2941 'job-queued': Job is in the 'processing' state, but more specifically, the Printer has queued the  
2942 document data.

2943 'job-transforming': Job is in the 'processing' state, but more specifically, the Printer is interpreting  
2944 document data and producing another electronic representation.

2945 'job-printing': The output device is marking media. This value is useful for Printers which spend a  
2946 great deal of time processing (1) when no marking is happening and then want to show that  
2947 marking is now happening or (2) when the job is in the process of being canceled or aborted while  
2948 the job remains in the 'processing' state, but the marking has not yet stopped so that impression or  
2949 sheet counts are still increasing for the job.

2950 'job-canceled-by-user': The job was canceled by the owner of the job using the Cancel-Job request,  
2951 i.e., by a user whose authenticated identity is the same as the value of the originating user that  
2952 created the Job object, or by some other authorized end-user, such as a member of the job owner's  
2953 security group.

2954 'job-canceled-by-operator': The job was canceled by the operator using the Cancel-Job request, i.e.,  
2955 by a user who has been authenticated as having operator privileges (whether local or remote). If  
2956 the security policy is to allow anyone to cancel anyone's job, then this value may be used when the  
2957 job is canceled by other than the owner of the job. For such a security policy, in effect, everyone  
2958 is an operator as far as canceling jobs with IPP is concerned.

2959 'job-canceled-at-device': The job was canceled by an unidentified local user, i.e., a user at a console  
2960 at the device.

2961 'aborted-by-system': The job (1) is in the process of being aborted, (2) has been aborted by the  
2962 system and placed in the 'aborted' state, or (3) has been aborted by the system and placed in the  
2963 'pending-held' state, so that a user or operator can manually try the job again.

2964 'processing-to-stop-point': The requester has issued a Cancel-job operation or the Printer object has  
2965 aborted the job, but is still performing some actions on the job until a specified stop point occurs  
2966 or job termination/cleanup is completed.

2967

2968 This reason is recommended to be used in conjunction with the 'processing' job state to indicate  
2969 that the Printer object is still performing some actions on the job while the job remains in the  
2970 'processing' state. After all the job's job description attributes have stopped incrementing, the  
2971 Printer object moves the job from the 'processing' state to the 'canceled' or 'aborted' job states.

2972

2973 'service-off-line': The Printer is off-line and accepting no jobs. All 'pending' jobs are put into the  
2974 'pending-held' state. This situation could be true if the service's or document transform's input is  
2975 impaired or broken.

2976 'job-completed-successfully': The job completed successfully.

2977 'job-completed-with-warnings': The job completed with warnings.

2978 'job-completed-with-errors': The job completed with errors (and possibly warnings too).

2979

#### 2980 4.3.9 job-state-message (text(MAX))

2981 This attribute specifies information about the "job-state" and "job-state-reasons" attributes in human  
2982 readable text. If the Printer object supports this attribute, the Printer object ~~SHALL~~**MUST** be able to  
2983 generate this message in any of the natural languages identified by the Printer's "generated-natural-  
2984 language-supported" attribute (see the "attributes-natural-language" operation attribute specified in  
2985 Section 3.1.4.1).

2986 Note: the value SHOULD NOT contain additional information not contained in the values of the "job-  
2987 state" and "job-states-reasons" attributes, such as interpreter error information. Otherwise, application  
2988 programs might attempt to parse the (localized text). For such additional information such as interpreter  
2989 errors for application program consumption, a new attribute with keyword values, needs to be developed  
2990 and registered.

#### 2991 4.3.10 number-of-documents (integer(0:MAX))

2992 This attribute indicates the number of documents in the job, i.e., the number of Send-Document, Send-  
2993 URI, Print-Job, or Print-URI operations that the Printer has accepted for this job, regardless of whether  
2994 the document data has reached the Printer object or not.

2995 Implementations supporting the OPTIONAL Create-Job/Send-Document/Send-URI operations  
2996 SHOULD support this attribute so that clients can query the number of documents in each job.

#### 2997 4.3.11 output-device-assigned (name(127))

2998 This attribute identifies the output device to which the Printer object has assigned this job. If an output  
2999 device implements an embedded Printer object, the Printer object NEED NOT set this attribute. If a print  
3000 server implements a Printer object, the value MAY be empty (zero-length string) or not returned until the  
3001 Printer object assigns an output device to the job. This attribute is particularly useful when a single  
3002 Printer object support multiple devices (so called "fan-out").

#### 3003 4.3.12 time-at-creation (integer(0:MAX))

3004 This attribute indicates the point in time at which the Job object was created. In order to populate this  
3005 attribute, the Printer object uses the value in its "printer-up-time" attribute at the time the Job object is  
3006 created.

#### 3007 4.3.13 time-at-processing (integer(0:MAX))

3008 This attribute indicates the point in time at which the Job object began processing. In order to populate  
3009 this attribute, the Printer object uses the value in its "printer-up-time" attribute at the time the Job object  
3010 is moved into the 'processing' state for the first time.

#### 3011 4.3.14 time-at-completed (integer(0:MAX))

3012 This attribute indicates the point in time at which the Job object completed (or was cancelled or aborted).  
3013 In order to populate this attribute, the Printer object uses the value in its "printer-up-time" attribute at the  
3014 time the Job object is moved into the 'completed' or 'canceled' or 'aborted' state.



## 3015 4.3.15 number-of-intervening-jobs (integer(0:MAX))

3016 This attribute indicates the number of jobs that are "ahead" of this job in the relative chronological order  
3017 of expected time to complete (i.e., the current scheduled order). For efficiency, it is only necessary to  
3018 calculate this value when an operation is performed that requests this attribute.

## 3019 4.3.16 job-message-from-operator (text(127))

3020 This attribute provides a message from an operator, system administrator or "intelligent" process to  
3021 indicate to the end user the reasons for modification or other management action taken on a job.

## 3022 4.3.17 job-k-octets (integer(0:MAX))

3023 This attribute specifies the total size of the document(s) in K octets, i.e., in units of 1024 octets requested  
3024 to be processed in the job. The value **SHALLMUST** be rounded up, so that a job between 1 and 1024  
3025 octets **SHALLMUST** be indicated as being 1, 1025 to 2048 **SHALLMUST** be 2, etc.

3026 This value **SHALLMUST** NOT include the multiplicative factors contributed by the number of copies  
3027 specified by the "copies" attribute, independent of whether the device can process multiple copies without  
3028 making multiple passes over the job or document data and independent of whether the output is collated  
3029 or not. Thus the value is independent of the implementation and indicates the size of the document(s)  
3030 measured in K octets independent of the number of copies.

3031 This value **SHALLMUST** also not include the multiplicative factor due to a copies instruction embedded  
3032 in the document data. If the document data actually includes replications of the document data, this value  
3033 will include such replication. In other words, this value is always the size of the source document data,  
3034 rather than a measure of the hardcopy output to be produced.

3035 Note: This attribute and the following two attributes ("job-impressions" and "job-media-sheets") are not  
3036 intended to be counters; they are intended to be useful routing and scheduling information if known. For  
3037 these three attributes, the Printer object may try to compute the value if it is not supplied in the create  
3038 request. Even if the client does supply a value for these three attributes in the create request, the Printer  
3039 object MAY choose to change the value if the Printer object is able to compute a value which is more  
3040 accurate than the client supplied value. The Printer object may be able to determine the correct value for  
3041 these three attributes either right at job submission time or at any later point in time.

## 3042 4.3.18 job-impressions (integer(0:MAX))

3043 This attribute specifies the total size in number of impressions of the document(s) being submitted (see  
3044 the definition of impression in section 13.2.5).



3045 As with "job-k-octets", this value **SHALLMUST** NOT include the multiplicative factors contributed by  
3046 the number of copies specified by the "copies" attribute, independent of whether the device can process  
3047 multiple copies without making multiple passes over the job or document data and independent of  
3048 whether the output is collated or not. Thus the value is independent of the implementation and reflects  
3049 the size of the document(s) measured in impressions independent of the number of copies.

3050 As with "job-k-octets", this value **SHALLMUST** also not include the multiplicative factor due to a copies  
3051 instruction embedded in the document data. If the document data actually includes replications of the  
3052 document data, this value will include such replication. In other words, this value is always the number of  
3053 impressions in the source document data, rather than a measure of the number of impressions to be  
3054 produced by the job.

3055 See the Note in the "job-k-octets" attribute that also applies to this attribute.

#### 3056 4.3.19 job-media-sheets (integer(0:MAX))

3057 This attribute specifies the total number of media sheets to be produced for this job.

3058 Unlike the "job-k-octets" and the "job-impressions" attributes, this value **SHALLMUST** include the  
3059 multiplicative factors contributed by the number of copies specified by the "copies" attribute and a  
3060 'number of copies' instruction embedded in the document data, if any. This difference allows the system  
3061 administrator to control the lower and upper bounds of both (1) the size of the document(s) with "job-k-  
3062 octets-supported" and "job-impressions-supported" and (2) the size of the job with "job-media-sheets-  
3063 supported".

3064 See the Note in the "job-k-octets" attribute that also applies to this attribute.

#### 3065 4.3.20 job-k-octets-processed (integer(0:MAX))

3066 This attribute specifies the total number of octets processed in K octets, i.e., in units of 1024 octets so  
3067 far. The value **SHALLMUST** be rounded up, so that a job between 1 and 1024 octets inclusive  
3068 **SHALLMUST** be indicated as being 1, 1025 to 2048 inclusive **SHALLMUST** be 2, etc.

3069 For implementations where multiple copies are produced by the interpreter with only a single pass over  
3070 the data, the final value **SHALLMUST** be equal to the value of the "job-k-octets" attribute. For  
3071 implementations where multiple copies are produced by the interpreter by processing the data for each  
3072 copy, the final value **SHALLMUST** be a multiple of the value of the "job-k-octets" attribute.

3073 Note: This attribute and the following two attributes ("job-impressions-completed" and "job-sheets-  
3074 completed") are intended to be counters. That is, the value for a job that has not started processing  
3075 **SHALLMUST** be 0. When the job's "job-state" is 'processing' or 'processing-stopped', this value is

3076 intended to contain the amount of the job that has been processed to the time at which the attributes are  
3077 requested.

#### 3078 4.3.21 job-impressions-completed (integer(0:MAX))

3079 This job attribute specifies the number of impressions completed for the job so far. For printing devices,  
3080 the impressions completed includes interpreting, marking, and stacking the output.

3081 See the note in "job-k-octets-processed" which also applies to this attribute.

#### 3082 4.3.22 job-media-sheets-completed (integer(0:MAX))

3083 This job attribute specifies the media-sheets completed marking and stacking for the entire job so far  
3084 whether those sheets have been processed on one side or on both.

3085 See the note in "job-k-octets-processed" which also applies to this attribute.

#### 3086 4.3.23 attributes-charset (charset)

3087 This REQUIRED attribute is populated using the value in the client supplied "attributes-charset" attribute  
3088 in the create request. It identifies the charset (coded character set and encoding method) used by any Job  
3089 attributes with attribute syntax 'text' and 'name' that were supplied by the client in the create request. See  
3090 Section 3.1.4 for a complete description of the "attributes-charset" operation attribute.

3091 This attribute does not indicate the charset in which the 'text' and 'name' values are stored internally in the  
3092 Job object. The internal charset is implementation-defined. The IPP object **SHALLMUST** convert from  
3093 whatever the internal charset is to that being requested in an operation as specified in Section 3.1.4.

#### 3094 4.3.24 attributes-natural-language (naturalLanguage)

3095 This REQUIRED attribute is populated using the value in the client supplied "attributes-natural-  
3096 language" attribute in the create request. It identifies the natural language used for any Job attributes  
3097 with attribute syntax 'text' and 'name' that were supplied by the client in the create request. See Section  
3098 3.1.4 for a complete description of the "attributes-natural-language" operation attribute. See Section  
3099 3.2.6 for how this attribute is returned in a Get-Jobs operation when jobs with different natural languages  
3100 are returned. See Sections 4.1.1.2 and 4.1.2.2 for how a Natural Language Override may be supplied  
3101 explicitly for each 'text' and 'name' attribute value that differs from the value identified by the "attributes-  
3102 natural-language" attribute.

## 3103 4.4 Printer Description Attributes

3104 These attributes form the attribute group called "printer-description". The following table summarizes  
3105 these attributes, their syntax, and whether or not they are REQUIRED for a Printer object to support. If  
3106 they are not indicated as REQUIRED, they are OPTIONAL. The maximum size in octets for 'text' and  
3107 'name' attributes is indicated in parentheses.

3108 Note: How these attributes are set by an Administrator is outside the scope of this specification.

3109	+	-----+	-----+	-----+
3110		Attribute		Syntax
3111		<u>MANDATORYREQUIRED?</u>		
3112	+	-----+	-----+	-----+
3113		printer-uri-supported		1setOf uri
3114		<u>MANDATORYREQUIRED</u>		
3115	+	-----+	-----+	-----+
3116		uri-security-supported		1setOf type2 keyword
3117		<u>MANDATORYREQUIRED</u>		
3118	+	-----+	-----+	-----+
3119		printer-name		name (127)
3120		<u>MANDATORYREQUIRED</u>		
3121	+	-----+	-----+	-----+
3122		printer-location		text (127)
3123	+	-----+	-----+	-----+
3124		printer-info		text (127)
3125	+	-----+	-----+	-----+
3126		printer-more-info		uri
3127	+	-----+	-----+	-----+
3128		printer-driver-installer		uri
3129	+	-----+	-----+	-----+
3130		printer-make-and-model		text (127)
3131	+	-----+	-----+	-----+
3132		printer-more-info-		uri
3133		manufacturer		
3134	+	-----+	-----+	-----+
3135		printer-state		type1 enum
3136		<u>MANDATORYREQUIRED</u>		
3137	+	-----+	-----+	-----+
3138		printer-state-reasons		1setOf type2 keyword
3139	+	-----+	-----+	-----+
3140		printer-state-message		text (MAX)
3141	+	-----+	-----+	-----+
3142		operations-supported		1setOf type2 enum
3143		<u>MANDATORYREQUIRED</u>		
3144	+	-----+	-----+	-----+
3145		charset-configured		charset
3146		<u>MANDATORYREQUIRED</u>		
3147	+	-----+	-----+	-----+
3148		charset-supported		1setOf charset
3149		<u>MANDATORYREQUIRED</u>		
3150	+	-----+	-----+	-----+
3151		natural-language-configured		naturalLanguage
3152		<u>MANDATORYREQUIRED</u>		
3153	+	-----+	-----+	-----+

3154	generated-natural-language-	1setOf	
3155	<u>MANDATORYREQUIRED</u>		
3156	supported	naturalLanguage	
3157	+-----+-----+-----+		
3158	document-format-default	mimeType	
3159	<u>MANDATORYREQUIRED</u>		
3160	+-----+-----+-----+		
3161	document-format-	1setOf	
3162	<u>MANDATORYREQUIRED</u>		
3163	supported	mimeType	
3164	+-----+-----+-----+		
3165	printer-is-accepting-jobs	boolean	
3166	<u>MANDATORYREQUIRED</u>		
3167	+-----+-----+-----+		
3168	queued-job-count	integer (0:MAX)	
3169	+-----+-----+-----+		
3170	printer-message-from-	text (127)	
3171	operator		
3172	+-----+-----+-----+		
3173	color-supported	boolean	
3174	+-----+-----+-----+		
3175	reference-uri-schemes-	1setOf uriScheme	
3176	supported		
3177	+-----+-----+-----+		
3178	pdl-override-supported	type2 keyword	
3179	<u>MANDATORYREQUIRED</u>		
3180	+-----+-----+-----+		
3181	printer-up-time	integer (1:MAX)	
3182	<u>MANDATORYREQUIRED</u>		
3183	+-----+-----+-----+		
3184	printer-current-time	dateTime	
3185	+-----+-----+-----+		
3186	multiple-operation-time-out	integer (1:MAX)	
3187	+-----+-----+-----+		
3188	compression-supported	1setOf type3 keyword	
3189	+-----+-----+-----+		
3190	job-k-octets-supported	rangeOfInteger	
3191		(0:MAX)	
3192	+-----+-----+-----+		
3193	job-impressions-supported	rangeOfInteger	
3194		(0:MAX)	
3195	+-----+-----+-----+		
3196	job-media-sheets-supported	rangeOfInteger	
3197		(0:MAX)	
3198	+-----+-----+-----+		

3199

## 3200 4.4.1 printer-uri-supported (1setOf uri)

3201 This **MANDATORYREQUIRED** Printer attribute contains at least one URI for the Printer object. It  
3202 OPTIONALLY contains more than one URI for the Printer object. An administrator determines a  
3203 Printer object's URI(s) and configures this attribute to contain those URIs by some means outside the  
3204 scope of IPP/1.0. The precise format of this URI is implementation dependent and depends on the  
3205 protocol. See the next section for a description "uri-security-supported" which is the  
3206 **MANDATORYREQUIRED** companion attribute to this "printer-uri-supported" attribute. See section  
3207 2.4 on Printer object identity and section 8.2 on security and URIs for more information.

## 3208 4.4.2 uri-security-supported (1setOf type2 keyword)

3209 This **MANDATORYREQUIRED** Printer attribute MUST have the same cardinality (contain the same  
3210 number of values) as the "printer-uri-supported" attribute. This attribute identifies the security  
3211 mechanisms used for each URI listed in the "printer-uri-supported" attribute. The "i th" value in "uri-  
3212 security-supported" corresponds to the "i th" value in "printer-uri-supported" and it describes the security  
3213 mechanisms used for accessing the Printer object via that URI. The following standard values are defined:

3214 'none': There are no secure communication channel protocols in use for the given URI.

3215 'tls': TLS 1.0 [TLS] is the secure communications channel protocol in use for the given URI.

3216 'ssl3': SSL3 is the secure communications channel protocol in use for the given URI.

3217

3218 Consider the following example. For a single Printer object, an administrator configures the "printer-uri-  
3219 supported" and "uri-security-supported" attributes as follows:

3220 "printer-uri-supported": 'http://acme.com/open-use-printer', 'http://acme.com/restricted-use-printer',

3221 'http://acme.com/private-printer'

3222 "uri-security-supported": 'none', 'none', 'tls'

3223

3224 In this case, one Printer object has three URIs.

3225 - For the first URI, 'http://acme.com/open-use-printer', the value 'none' in "uri-security-supported"  
3226 indicates that there is no secure channel protocol configured to run under HTTP. The name  
3227 implies that there is no Basic or Digest authentication being used, but it is up to the client to  
3228 determine that while using HTTP underneath the IPP application protocol.

3229 - For the second URI, 'http://acme.com/restricted-use-printer', the value 'none' in "uri-security-  
3230 supported" indicates that there is no secure channel protocol configured to run under HTTP. In  
3231 this case, although the name does imply that there is some sort of Basic or Digest authentication

3232 being used within HTTP, it is up to the client to determine that while using HTTP and by  
3233 processing any '401 Unauthorized' HTTP error messages.  
3234 - For the third URI, 'http://acme.com/private-printer', the value 'tls' in "uri-security-supported"  
3235 indicates that TLS is being used to secure the channel. The client SHOULD be prepared to use  
3236 TLS framing to negotiate an acceptable ciphersuite to use while communicating with the Printer  
3237 object. In this case, the name implies the use of a secure communications channel, but the fact is  
3238 made explicit by the presence of the 'tls' value in "uri-security-supported". The client does not  
3239 need to resort to understanding which security it must use by following naming conventions or by  
3240 parsing the URI to determine which security mechanisms are implied.

3241

3242 It is expected that many IPP Printer objects will be configured to support only one channel (either  
3243 configured to use TLS access or not), and will therefore only ever have one URI listed in the "printer-uri-  
3244 supported" attribute. No matter the configuration of the Printer object (whether it has only one URI or  
3245 more than one URI), a client ~~SHALL~~**MUST** supply only one URI in the target "printer-uri" operation  
3246 attribute.

#### 3247 4.4.3 printer-name (name(127))

3248 This ~~MANDATORYREQUIRED~~ Printer attribute contains the name of the Printer object. It is a name  
3249 that is more end-user friendly than a URI. An administrator determines a printer's name and sets this  
3250 attribute to that name. This name may be the last part of the printer's URI or it may be unrelated. In non-  
3251 US-English locales, a name may contain characters that are not allowed in a URI.

#### 3252 4.4.4 printer-location (text(127))

3253 This Printer attribute identifies the location of the device. This could include things like: "in Room 123A,  
3254 second floor of building XYZ".

#### 3255 4.4.5 printer-info (text(127))

3256 This Printer attribute identifies the descriptive information about this Printer object. This could include  
3257 things like: "This printer can be used for printing color transparencies for HR presentations", or "Out of  
3258 courtesy for others, please print only small (1-5 page) jobs at this printer", or even "This printer is going  
3259 away on July 1, 1997, please find a new printer".

#### 3260 4.4.6 printer-more-info (uri)

3261 This Printer attribute contains a URI used to obtain more information about this specific Printer object.  
3262 For example, this could be an HTTP type URI referencing an HTML page accessible to a Web Browser.  
3263 The information obtained from this URI is intended for end user consumption. Features outside the scope

3264 of IPP can be accessed from this URI. The information is intended to be specific to this printer instance  
 3265 and site specific services (e.g. job pricing, services offered, end user assistance). The device manufacturer  
 3266 may initially populate this attribute.

#### 3267 4.4.7 printer-driver-installer (uri)

3268 This Printer attribute contains a URI to use to locate the driver installer for this Printer object. This  
 3269 attribute is intended for consumption by automata. The mechanics of print driver installation is outside  
 3270 the scope of IPP. The device manufacturer may initially populate this attribute.

#### 3271 4.4.8 printer-make-and-model (text(127))

3272 This Printer attribute identifies the make and model of the device. The device manufacturer may initially  
 3273 populate this attribute.

#### 3274 4.4.9 printer-more-info-manufacturer (uri)

3275 This Printer attribute contains a URI used to obtain more information about this type of device. The  
 3276 information obtained from this URI is intended for end user consumption. Features outside the scope of  
 3277 IPP can be accessed from this URI (e.g., latest firmware, upgrades, print drivers, optional features  
 3278 available, details on color support). The information is intended to be germane to this printer without  
 3279 regard to site specific modifications or services. The device manufacturer may initially populate this  
 3280 attribute.

#### 3281 4.4.10 printer-state (type1 enum)

3282 This **MANDATORYREQUIRED** Printer attribute identifies the current state of the device. The "printer-  
 3283 state reasons" attribute augments the "printer-state" attribute to give more detailed information about the  
 3284 Printer in the given printer state.

3285 A Printer object need only update this attribute before responding to an operation which requests the  
 3286 attribute; the Printer object NEED NOT update this attribute continually, since asynchronous event  
 3287 notification is not part of IPP/1.0. A Printer NEED NOT implement all values if they are not applicable  
 3288 to a given implementation.

3289 The following standard values are defined:

3290	Value	Symbolic Name and Description
3291		



- 3292 '3' 'idle': If a Printer receives a job (whose required resources are ready) while in this state,  
3293 such a job **SHALLMUST** transit into the processing state immediately. If the  
3294 printer-state-reasons attribute contains any reasons, they **SHALLMUST** be  
3295 reasons that would not prevent a job from transiting into the processing state  
3296 immediately, e.g., toner-low. Note: if a Printer controls more than one output  
3297 device, the above definition implies that a Printer is idle if at least one output  
3298 device is idle.  
3299
- 3300 '4' 'processing': If a Printer receives a job (whose required resources are ready) while in this  
3301 state, such a job **SHALLMUST** transit into the pending state immediately. Such a  
3302 job **SHALLMUST** transit into the processing state only after jobs ahead of it  
3303 complete. If the printer-state-reasons attribute contains any reasons, they  
3304 **SHALLMUST** be reasons that do not prevent the current job from printing, e.g.  
3305 toner-low. Note: if a Printer controls more than one output device, the above  
3306 definition implies that a Printer is processing if at least one output device is  
3307 processing, and none is idle.  
3308
- 3309 '5' 'stopped': If a Printer receives a job (whose required resources are ready) while in this  
3310 state, such a job **SHALLMUST** transit into the pending state immediately. Such a  
3311 job **SHALLMUST** transit into the processing state only after some human fixes the  
3312 problem that stopped the printer and after jobs ahead of it complete printing. If  
3313 supported, the "printer-state-reasons" attribute **SHALLMUST** contain at least one  
3314 reason, e.g. media-jam, which prevents it from either processing the current job or  
3315 transitioning a pending job to the processing state.  
3316
- 3317 Note: if a Printer controls more than one output device, the above definition  
3318 implies that a Printer is stopped only if all output devices are stopped. Also, it is  
3319 tempting to define stopped as when a sufficient number of output devices are  
3320 stopped and leave it to an implementation to define the sufficient number. But  
3321 such a rule complicates the definition of stopped and processing. For example,  
3322 with this alternate definition of stopped, a job can move from idle to processing  
3323 without human intervention, even though the Printer is stopped.  
3324

#### 3325 4.4.11 printer-state-reasons (1setOf type2 keyword)

3326 This Printer attribute supplies additional detail about the device's state.

3327 Each keyword value MAY have a suffix to indicate its level of severity. The three levels are: report (least  
3328 severe), warning, and error (most severe).

- 3329 - '-report': This suffix indicates that the reason is a "report". An implementation may choose to omit  
3330 some or all reports. Some reports specify finer granularity about the printer state; others serve as  
3331 a precursor to a warning. A report **SHALLMUST** contain nothing that could affect the printed  
3332 output.
- 3333 - '-warning': This suffix indicates that the reason is a "warning". An implementation may choose to  
3334 omit some or all warnings. Warnings serve as a precursor to an error. A warning **SHALLMUST**  
3335 contain nothing that prevents a job from completing, though in some cases the output may be of  
3336 lower quality.
- 3337 - '-error': This suffix indicates that the reason is an "error". An implementation **SHALLMUST**  
3338 include all errors. If this attribute contains one or more errors, printer **SHALLMUST** be in the  
3339 stopped state.

3340

3341 If the implementation does not add any one of the three suffixes, all parties **SHALLMUST** assume that  
3342 the reason is an "error".

3343 If a Printer object controls more than one output device, each value of this attribute MAY apply to one or  
3344 more of the output devices. An error on one output device that does not stop the Printer object as a  
3345 whole MAY appear as a warning in the Printer's "printer-state-reasons attribute". If the "printer-state"  
3346 for such a Printer has a value of 'stopped', then there MUST be an error reason among the values in the  
3347 "printer-state-reasons" attribute.

3348 The following standard values are defined:

3349 'other': The device has detected an error other than one listed in this document.

3350 'none': There are not reasons. This state reason is semantically equivalent to "printer-state-reasons"  
3351 without any value.

3352 'media-needed': A tray has run out of media.

3353 'media-jam': The device has a media jam.

3354 'paused': Someone has paused the Printer object. In this state, a Printer **SHALLMUST** NOT produce  
3355 printed output, but it **SHALLMUST** perform other operations requested by a client. If a Printer  
3356 had been printing a job when the Printer was paused, the Printer **SHALLMUST** resume printing  
3357 that job when the Printer is no longer paused and leave no evidence in the printed output of such a  
3358 pause.

3359 'shutdown': Someone has removed a Printer object from service, and the device may be powered  
3360 down or physically removed. In this state, a Printer object **SHALLMUST** NOT produce printed  
3361 output, and unless the Printer object is realized by a print server that is still active, the Printer  
3362 object **SHALLMUST** perform no other operations requested by a client, including returning this  
3363 value. If a Printer object had been printing a job when it was shutdown, the Printer ~~need~~  
3364 ~~not~~**NEED NOT** resume printing that job when the Printer is no longer shutdown. If the Printer  
3365 resumes printing such a job, it may leave evidence in the printed output of such a shutdown, e.g.  
3366 the part printed before the shutdown may be printed a second time after the shutdown.

3367 'connecting-to-device': The Printer object has scheduled a job on the output device and is in the  
3368 process of connecting to a shared network output device (and might not be able to actually start  
3369 printing the job for an arbitrarily long time depending on the usage of the output device by other  
3370 servers on the network).

3371 'timed-out': The server was able to connect to the output device (or is always connected), but was  
3372 unable to get a response from the output device.

3373 'stopping': The Printer object is in the process of stopping the device and will be stopped in a while.  
3374 When the device is stopped, the Printer object will change the Printer object's state to 'stopped'.  
3375 The 'stopping-warning' reason is never an error, even for a Printer with a single output device.  
3376 When an output-device ceases accepting jobs, the Printer will have this reason while the output  
3377 device completes printing.

3378 'stopped-partly': When a Printer object controls more than one output device, this reason indicates  
3379 that one or more output devices are stopped. If the reason is a report, fewer than half of the  
3380 output devices are stopped. If the reason is a warning, fewer than all of the output devices are  
3381 stopped.

3382 'toner-low': The device is low on toner.

3383 'marker-supply-low': The device is low on marker supply (ink, paint, etc.).

3384 'spool-area-full': The limit of persistent storage allocated for spooling has been reached.

3385 'cover-open': One or more covers on the device are open.

3386 'interlock-open': One or more interlock devices on the printer are unlocked.

3387 'door-open': One or more doors on the device are open.

3388 'input-tray-missing': One or more input trays are not in the device.

3389 'media-low': At least one input tray is low on media.

3390 'media-empty': At least one input tray is empty.

3391 'output-tray-missing': One or more output trays are not in the device

3392 'output-area-almost-full': One or more output area is almost full (e.g. tray, stacker, collator).

3393 'output-area-full': One or more output area is full. (e.g. tray, stacker, collator)

3394 'marker-supply-low': The device is low on at least one marker supply. (e.g. toner, ink, ribbon)

3395 'marker-supply-empty': The device is out of at least one marker supply. (e.g. toner, ink, ribbon)

3396 'marker-waste-almost-full': The device marker supply waste receptacle is almost full.

3397 'marker-waste-full': The device marker supply waste receptacle is full.

3398 'fuser-over-temp': The fuser temperature is above normal.

3399 'fuser-under-temp': The fuser temperature is below normal.

3400 'opc-near-eol': The optical photo conductor is near end of life.

3401 'opc-life-over': The optical photo conductor is no longer functioning.

3402 'developer-low': The device is low on developer.

3403 'developer-empty': The device is out of developer.

3404 'interpreter-resource-unavailable': An interpreter resource is unavailable (i.e. font, form)

3405

## 3406 4.4.12 printer-state-message (text(MAX))

3407 This Printer attribute specifies the additional information about the printer state and printer state reasons  
 3408 in human readable text. If the Printer object supports this attribute, the Printer object **SHALLMUST** be  
 3409 able to generate this message in any of the natural languages identified by the Printer's "generated-natural-  
 3410 language-supported" attribute (see the "attributes-natural-language" operation attribute specified in  
 3411 Section 3.1.4.1).

## 3412 4.4.13 operations-supported (1setOf type2 enum)

3413 This **MANDATORYREQUIRED** Printer attribute specifies the set of supported operations for this  
 3414 Printer object and contained Job objects. No 32-bit enum value for this attribute **SHALLMUST** exceed  
 3415 0x8FFF, since these values are passed in two octets in each Protocol request [IPP-PRO].

3416 The following standard values are defined:

3417	Value	Operation Name
3418	-----	-----
3419		
3420	0x0000	reserved, not used
3421	0x0001	reserved, not used
3422	0x0002	Print-Job
3423	0x0003	Print-URI
3424	0x0004	Validate-Job
3425	0x0005	Create-Job
3426	0x0006	Send-Document
3427	0x0007	Send-URI
3428	0x0008	Cancel-Job
3429	0x0009	Get-Job-Attributes
3430	0x000A	Get-Jobs
3431	0x000B	Get-Printer-Attributes
3432	0x000C-0x3FFF	reserved for future operations
3433	0x4000-0x8FFF	reserved for private extensions

3434  
 3435 This allows for certain vendors to implement private extensions that are guaranteed to not conflict with  
 3436 future registered extensions. However, there is no guarantee that two or more private extensions will not  
 3437 conflict.

## 3438 4.4.14 charset-configured (charset)

3439 This **MANDATORYREQUIRED** Printer attribute identifies the charset that the Printer object has been  
3440 configured to represent 'text' and 'name' Printer attributes that are set by the operator, system  
3441 administrator, or manufacturer, i.e., for "printer-name" (name), "printer-location" (text), "printer-info"  
3442 (text), and "printer-make-and-model" (text). Therefore, the value of the Printer object's "charset-  
3443 configured" attribute **SHALLMUST** also be among the values of the Printer object's "charset-supported"  
3444 attribute.

## 3445 4.4.15 charset-supported (1setOf charset)

3446 This **MANDATORYREQUIRED** Printer attribute identifies the set of charsets that the Printer and  
3447 contained Job objects support in attributes with attribute syntax 'text' and 'name'. At least the value 'utf-8'  
3448 **SHALLMUST** be present, since IPP objects **MUST** support the UTF-8 [RFC2044] charset. If a Printer  
3449 object supports a charset, it means that for all attributes of syntax 'text' and 'name' the IPP object  
3450 **SHALLMUST** (1) accept the charset in requests and return the charset in responses as needed.

3451 If more charsets than UTF-8 are supported, the IPP object **SHALLMUST** perform charset conversion  
3452 between the charsets as described in Section 3.2.1.2.

## 3453 4.4.16 natural-language-configured (naturalLanguage)

3454 This **MANDATORYREQUIRED** Printer attribute identifies the natural language that the Printer object  
3455 has been configured to represent 'text' and 'name' Printer attributes that are set by the operator, system  
3456 administrator, or manufacturer, i.e., for "printer-name" (name), "printer-location" (text), "printer-info"  
3457 (text), and "printer-make-and-model" (text). When returning these Printer attributes, the Printer object  
3458 **MAY** return them in the configured natural language specified by this attribute, instead of the natural  
3459 language requested by the client in the "attributes-natural-language" operation attribute. See Section  
3460 3.1.4.1 for the specification of the **OPTIONAL** multiple natural language support. Therefore, the value  
3461 of the Printer object's "natural-language-configured" attribute **SHALLMUST** also be among the values of  
3462 the Printer object's "generated-natural-language-supported" attribute.

## 3463 4.4.17 generated-natural-language-supported (1setOf naturalLanguage)

3464 This **MANDATORYREQUIRED** Printer attribute identifies the natural language(s) that the Printer  
3465 object and contained Job objects support in attributes with attribute syntax 'text' and 'name'. The natural  
3466 language(s) supported depends on implementation and/or configuration. Unlike charsets, IPP objects  
3467 **SHALLMUST** accept requests with any natural language or any Natural Language Override whether the  
3468 natural language is supported or not.

3469 If a Printer object supports a natural language, it means that for any of the attributes for which the Printer  
3470 or Job object generates messages, i.e., for the "job-state-message" and "printer-state-message" attributes  
3471 and Operation Messages (see Section 3.1.5) in operation responses, the Printer and Job objects  
3472 **SHALLMUST** be able to generate messages in any of the Printer's supported natural languages. See  
3473 section 3.1.4 for the specification of 'text' and 'name' attributes in operation requests and responses.

3474 Note: A Printer object that supports multiple natural languages, often has separate catalogs of messages,  
3475 one for each natural language supported.

#### 3476 4.4.18 document-format-default (mimeMediaType)

3477 This Printer attribute identifies the document format that the Printer object has been configured to assume  
3478 if the client does not supply a "document-format" operation attribute in any of the operation requests that  
3479 supply document data. The standard values for this attribute are Internet Media types (sometimes called  
3480 MIME types). For further details see the description of the 'mimeMediaType' attribute syntax in Section  
3481 4.1.9.

#### 3482 4.4.19 document-format-supported (1setOf mimeMediaType)

3483 This Printer attribute identifies the set of document formats that the Printer object and contained Job  
3484 objects can support. For further details see the description of the 'mimeMediaType' attribute syntax in  
3485 Section 4.1.9.

#### 3486 4.4.20 printer-is-accepting-jobs (boolean)

3487 This **MANDATORYREQUIRED** Printer attribute indicates whether the printer is currently able to  
3488 accept jobs, i.e., is accepting Print-Job, Print-URI, and Create-Job requests. If the value is 'true', the  
3489 printer is accepting jobs. If the value is 'false', the Printer object is currently rejecting any jobs submitted  
3490 to it. In this case, the Printer object returns the 'server-error-not-accepting-jobs' status code.

3491 Note: This value is independent of the "printer-state" and "printer-state-reasons" attributes because its  
3492 value does not affect the current job; rather it affects future jobs. This attribute may cause the Printer to  
3493 reject jobs when the "printer-state" is 'idle' or it may cause the Printer object to accept jobs when the  
3494 "printer-state" is 'stopped'.

#### 3495 4.4.21 queued-job-count (integer(0:MAX))

3496 This Printer attribute contains a count of the number of jobs that are either 'pending', 'processing',  
3497 'pending-held', or 'processing-stopped' and is set by the Printer object.

## 3498 4.4.22 printer-message-from-operator (text(127))

3499 This Printer attribute provides a message from an operator, system administrator or "intelligent" process  
3500 to indicate to the end user information or status of the printer, such as why it is unavailable or when it is  
3501 expected to be available.

## 3502 4.4.23 color-supported (boolean)

3503 This Printer attribute identifies whether the device is capable of any type of color printing at all, including  
3504 highlight color. All document instructions having to do with color are embedded within the document  
3505 PDL (none are external IPP attributes in IPP/1.0).

3506 Note: end-users are able to determine the nature and details of the color support by querying the  
3507 "printer-more-info-manufacturer" Printer attribute.

## 3508 4.4.24 reference-uri-schemes-supported (1setOf uriScheme)

3509 This Printer attribute specifies which URI schemes are supported for use in the "document-uri" operation  
3510 attribute of the Print-URI or Send-URI operation. If a Printer object supports these optional operations,  
3511 it MUST support the "reference-uri-schemes-supported" Printer attribute with at least the following  
3512 schemed URI values:

3513 'ftp': The Printer object will use an FTP 'get' operation as defined in RFC 2228 [RFC2228] using FTP  
3514 URLs [as defined by](#) [RFC1738] [and](#) [RFC2316].

3515  
3516 The Printer object MAY OPTIONALLY support other URI schemes (see section 4.1.6).

## 3517 4.4.25 pdl-override-supported (type2 keyword)

3518 This **MANDATORYREQUIRED** Printer attribute expresses the ability for a particular Printer  
3519 implementation to either attempt to override document data instructions with IPP attributes or not.

3520 This attribute takes on the following values:

- 3521 - 'attempted': This value indicates that the Printer object attempts to make the IPP attribute values  
3522 take precedence over embedded instructions in the document data, however there is no guarantee.
- 3523 - 'not-attempted': This value indicates that the Printer object makes no attempt to make the IPP  
3524 attribute values take precedence over embedded instructions in the document data.

3525  
3526 Section 16 contains a full description of how this attribute interacts with and affects other IPP attributes,  
3527 especially the "ipp-attribute-fidelity" attribute.



## 3528 4.4.26 printer-up-time (integer(1:MAX))

3529 This **MANDATORYREQUIRED** Printer attribute indicates the amount of time (in seconds) that this  
3530 instance of this Printer implementation has been up and running. This value is used to populate the Job  
3531 attributes "time-at-creation", "time-at-processing", and "time-at-completed". These time values are all  
3532 measured in seconds and all have meaning only relative to this attribute, "printer-up-time". The value is a  
3533 monotonically increasing value starting from 1 when the Printer object is started-up (initialized, booted,  
3534 etc.).

3535 If the Printer object goes down at some value 'n', and comes back up, the implementation MAY:

- 3536 1. Know how long it has been down, and resume at some value greater than 'n', or
- 3537 2. Restart from 1.

3538

3539 In the first case, the Printer SHOULD not tweak any existing related Job attributes ("time-at-creation",  
3540 "time-at-processing", and "time-at-completed"). In the second case, the Printer object SHOULD reset  
3541 those attributes to 0. If a client queries a time-related Job attribute and finds the value to be 0, the client  
3542 MUST assume that the Job was submitted in some life other than the Printer's current life.

## 3543 4.4.27 printer-current-time (dateTime)

3544 This Printer attribute indicates the current absolute wall-clock time. If an implementation supports this  
3545 attribute, then a client could calculate the absolute wall-clock time each Job's "time-at-creation", "time-at-  
3546 processing", and "time-at-completed" attributes by using both "printer-up-time" and this attribute,  
3547 "printer-current-time". If an implementation does not support this attribute, a client can only calculate  
3548 the relative time of certain events based on the **MANDATORYREQUIRED** "printer-up-time" attribute.

## 3549 4.4.28 multiple-operation-time-out (integer(1:MAX))

3550 This Printer attributes identifies how long (in seconds) the Printer object waits for additional Send-  
3551 Document or Send-URI operations to follow a still-open multi-document Job object before taking one of  
3552 the actions indicated in section 3.3.1.

## 3553 4.4.29 compression-supported (1setOf type3 keyword)

3554 This Printer attribute identifies the set of supported compression algorithms for document data.  
3555 Compression only applies to the document data; compression does not apply to the encoding of the IPP  
3556 operation itself. The supported values are used to validate the client supplied "compression" operation  
3557 attributes in Print-Job, Send-Document, and Send-URI requests.

3558 Standard values are :



3559 'none': no compression is used.  
3560 'deflate': ZIP public domain inflate/deflate) compression technology  
3561 'gzip' GNU zip compression technology described in RFC 1952 [RFC1952].  
3562 'compress': UNIX compression technology  
3563

#### 3564 4.4.30 job-k-octets-supported (rangeOfInteger(0:MAX))

3565 This Printer attribute specifies the upper and lower bounds of total sizes of jobs in K octets, i.e., in units  
3566 of 1024 octets. The supported values are used to validate the client supplied "job-k-octets" operation  
3567 attributes in create requests. The corresponding job description attribute "job-k-octets" is defined in  
3568 section 4.3.17.

#### 3569 4.4.31 job-impressions-supported (rangeOfInteger(0:MAX))

3570 This Printer attribute specifies the upper and lower bounds for the number of impressions per job. The  
3571 supported values are used to validate the client supplied "job-impressions" operation attributes in create  
3572 requests. The corresponding job description attribute "job-impressions" is defined in section 4.3.18.

#### 3573 4.4.32 job-media-sheets-supported (rangeOfInteger(0:MAX))

3574 This Printer attribute specifies the upper and lower bounds for the number of media sheets per job. The  
3575 supported values are used to validate the client supplied "job-media-sheets" operation attributes in create  
3576 requests. The corresponding Job attribute "job-media-sheets" is defined in section 4.3.19.

## 3577 5. Conformance

3578 This section describes conformance issues and requirements. This document introduces model entities  
3579 such as objects, operations, attributes, attribute syntaxes, and attribute values. These conformance  
3580 sections describe the conformance requirements which apply to these model entities.

### 3581 5.1 Client Conformance Requirements

3582 A conforming client **SHALLMUST** support all **MANDATORYREQUIRED** operations as defined in this  
3583 document. For each attribute included in an operation request, a conforming client **SHALLMUST** supply  
3584 a value whose type and value syntax conforms to the requirements of the Model document as specified in  
3585 Sections 3 and 4. A conforming client MAY supply any registered extensions and/or private extensions  
3586 in an operation request, as long as they meet the requirements in Section 6.

3587 Otherwise, there are no conformance requirements placed on the user interfaces provided by IPP clients  
3588 or their applications. For example, one application might not allow an end user to submit multiple  
3589 documents per job, while another does. One application might first query a Printer object in order to  
3590 supply a graphical user interface (GUI) dialogue box with supported and default values whereas a  
3591 different implementation might not.

3592 When sending a request, an IPP client NEED NOT supply any attributes that are indicated as  
3593 OPTIONALLY supplied by the client.

3594 A client ~~SHALL~~**MUST** be able to accept any of the attribute syntaxes defined in Section 4.1, including  
3595 their full range, that may be returned to it in a response from a Printer object. For presentation purposes,  
3596 truncation of long attribute values is not recommended. A recommended approach would be for the  
3597 client implementation to allow the user to scroll through long attribute values.

3598 A query response may contain attribute groups, attributes, and values that the client does not expect.  
3599 Therefore, a client implementation **MUST** gracefully handle such responses and not refuse to inter-  
3600 operate with a conforming Printer that is returning extended registered or private attributes and/or  
3601 attribute values that conform to Section 6. Clients may choose to ignore any parameters, attributes, or  
3602 values that they do not understand.

## 3603 5.2 IPP Object Conformance Requirements

3604 This section specifies the conformance requirements for conforming implementations with respect to  
3605 objects, operations, and attributes.

### 3606 5.2.1 Objects

3607 Conforming implementations ~~SHALL~~**MUST** implement all of the model objects as defined in this  
3608 specification in the indicated sections:

3609 Section 2.1 - Printer Object

3610 Section 2.2 - Job Object

3611

### 3612 5.2.2 Operations

3613 Conforming IPP object implementations ~~SHALL~~**MUST** implement all of the  
3614 ~~MANDATORYREQUIRED~~ model operations, including ~~mandatory~~**REQUIRED** responses, as defined in  
3615 this specification in the indicated sections:

3616 For a Printer object:

3617	Print-Job (section 3.2.1)	<del>MANDATORY</del> <u>REQUIRED</u>
3618	Print-URI (section 3.2.2)	OPTIONAL
3619	Validate-Job (section 3.2.3)	<del>MANDATORY</del> <u>REQUIRED</u>
3620	Create-Job (section 3.2.4)	OPTIONAL
3621	Get-Printer-Attributes (section 3.2.5)	<del>MANDATORY</del> <u>REQUIRED</u>
3622	Get-Jobs (section 3.2.6)	<del>MANDATORY</del> <u>REQUIRED</u>
3623		
3624	For a Job object:	
3625	Send-Document (section 3.3.1)	OPTIONAL
3626	Send-URI (section 3.3.2)	OPTIONAL
3627	Cancel-Job (section 3.3.3)	<del>MANDATORY</del> <u>REQUIRED</u>
3628	Get-Job-Attributes (section 3.3.4)	<del>MANDATORY</del> <u>REQUIRED</u>

3629  
 3630 Conforming IPP objects ~~SHALL~~MUST support all ~~MANDATORY~~REQUIRED operation attributes and  
 3631 all values of such attributes if so indicated in the description. Conforming IPP objects ~~SHALL~~MUST  
 3632 ignore all unsupported or unknown operation attributes or operation attribute groups received in a  
 3633 request, but ~~SHALL~~MUST reject a request that contains a supported operation attribute that contains an  
 3634 unsupported value.

3635 The following section on object attributes specifies the support required for object attributes.

### 3636 5.2.3 IPP Object Attributes

3637 Conforming IPP objects ~~SHALL~~MUST support all of the ~~MANDATORY~~REQUIRED object attributes,  
 3638 as defined in this specification in the indicated sections.

3639 If an object supports an attribute, it ~~SHALL~~MUST support only those values specified in this document  
 3640 or through the extension mechanism described in section 5.2.4. It MAY support any non-empty subset of  
 3641 these values. That is, it ~~SHALL~~MUST support at least one of the specified values and at most all of  
 3642 them.

### 3643 5.2.4 Extensions

3644 A conforming IPP object MAY support registered extensions and private extensions, as long as they meet  
 3645 the requirements specified in Section 6.

3646 For each attribute included in an operation response, a conforming IPP object ~~SHALL~~MUST return a  
 3647 value whose type and value syntax conforms to the requirement of the Model document as specified in  
 3648 Sections 3 and 4.

3649 5.2.5 Attribute Syntaxes

3650 An IPP object **SHALLMUST** be able to accept any of the attribute syntaxes defined in Section 4.1,  
3651 including their full range, in any operation in which a client may supply attributes or the system  
3652 administrator may configure attributes (by means outside the scope of IPP/1.0). Furthermore, an IPP  
3653 object **SHALLMUST** return attributes to the client in operation responses that conform to the syntax  
3654 specified in Section 4.1, including their full range if supplied previously by a client.

3655 5.3 Charset and Natural Language Requirements

3656 All clients and IPP objects **SHALLMUST** support the 'utf-8' charset as defined in section 4.1.7.

3657 IPP objects **MUST** be able to accept any client request which correctly uses the "attributes-natural-  
3658 language" operation attribute or the Natural Language Override mechanism on any individual attribute  
3659 whether or not the natural language is supported by the IPP object. If an IPP object supports a natural  
3660 language, then it **MUST** be able to translate (perhaps by table lookup) all generated 'text' or 'name'  
3661 attribute values into one of the supported languages (see section 3.1.4). That is, the IPP object that  
3662 supports a natural language **NEED NOT** be a general purpose translator of any arbitrary 'text' or 'name'  
3663 value supplied by the client into that natural language. However, the object **MUST** be able to translate  
3664 (automatically generate) any of its own attribute values and messages into that natural language.

3665 5.4 Security Conformance Requirements

3666 Conforming IPP Printer objects **MAY** support Transport Layer Security (TLS) access, support access  
3667 without TLS or support both means of access.

3668 Conforming IPP clients **SHOULD** support TLS access and non-TLS access. Note: This client  
3669 requirement to support both means that conforming IPP clients will be able to inter-operate with any IPP  
3670 Printer object.

3671 For a detailed discussion of security considerations and the IPP application security profile required for  
3672 TLS support, see section 8.

3673 6. IANA Considerations (registered and private extensions)

3674 This section describes how IPP can be extended to allow the following registered and private extensions  
3675 to IPP:

3676 1. keyword attribute values

- 3677 2. enum attribute values
- 3678 3. attributes
- 3679 4. attribute syntaxes
- 3680 5. operations
- 3681 6. status codes

3682

3683 ~~Registered and private e~~Extensions registered for use with IPP/1.0 are OPTIONAL for client and IPP  
3684 object conformance to the IPP/1.0 Model specification.

3685 These extension procedures are aligned with the guidelines as set forth by the IESG [IANA-CON].  
3686 Section 12 describes how to propose new registrations for consideration. IANA will reject registration  
3687 proposals that leave out required information or do not follow the appropriate format described in  
3688 Section 12. IPP/1.0 may also be extended by an appropriate RFC that specifies any of the above  
3689 extensions.

## 3690 6.1 Typed 'keyword' and 'enum' Extensions

3691 IPP allows for 'keyword' and 'enum' extensions (see sections 4.1.3 and 4.1.4). This document uses  
3692 prefixes to the 'keyword' and 'enum' basic attribute syntax type in order to communicate extra information  
3693 to the reader through its name. This extra information ~~need not be~~ is not represented in the protocol  
3694 because it is unimportant to a client or Printer object. The list below describes the prefixes and their  
3695 meaning.

3696 "type1": The IPP specification must be revised to add a new keyword or a new enum. No private  
3697 keywords or enums are allowed.

3698

3699 "type2": Implementers can, at any time, add new keyword or enum values by proposing the complete  
3700 specification to IANA:

3701

3702 iana@iana.org

3703

3704 IANA will forward the registration proposal to the IPP Designated Expert who will review the  
3705 proposal with a mailing list that the Designated Expert keeps for this purpose. Initially, that list  
3706 will be the mailing list used by the IPP WG:

3707

3708 ipp@pwg.org

3709

3710 even after the IPP WG is disbanded as permitted by [IANA-CON]. The IPP Designated Expert is  
3711 appointed by the IESG Area Director responsible for IPP, according to [IANA-CON].

3712

3713           When a type2 keyword or enum is approved, the IPP Designated Expert becomes the point of  
3714           contact for any future maintenance that might be required for that registration.

3715  
3716           "type3": Implementers can, at any time, add new keyword and enum values by submitting the  
3717           complete specification to IANA as for type2 who will forward the proposal to the IPP Designated  
3718           Expert. While no additional technical review is required, the IPP Designated Expert may, at  
3719           his/her discretion, forward the proposal to the same mailing list as for type2 registrations for  
3720           advice and comment.

3721  
3722           When a type3 keyword or enum is approved by the IPP Designated Expert, the original proposer  
3723           becomes the point of contact for any future maintenance that might be required for that  
3724           registration.

3725  
3726           For type2 and type3 keywords, the proposer includes the name of the keyword in the registration  
3727           proposal and the name is part of the technical review.

3728           After type2 and type3 enums specifications are approved, the IPP Designated Expert in consultation with  
3729           IANA assigns the next available enum number for each enum value.

3730           IANA will publish approved type2 and type3 keyword and enum attributes value registration  
3731           specifications in:

3732                 ftp.isi.edu/iana/assignments/ipp/attribute-values/xxx/yyy.txt

3733           where xxx is the attribute name that specifies the initial values and yyy.txt is a descriptive file name that  
3734           contains one or more enums or keywords approved at the same time. For example, if several additional  
3735           enums for stapling are approved for use with the "finishings" attribute (and "finishings-default" and  
3736           "finishings-supported" attributes), IANA will publish the additional values in the file:

3737                 ftp.isi.edu/iana/assignments/ipp/attribute-values/finishings/stapling.txt.

3738           Note: Some attributes are defined to be: ~~either~~ 'type3 keywords' ~~and~~ 'name' which allows for attribute  
3739           values to be extended by a site administrator with administrator defined names. Such names are not  
3740           registered with IANA.

3741           By definition, each of the three types above assert some sort of registry or review process in order for  
3742           extensions to be considered valid. Each higher numbered level (1, 2, 3) tends to be decreasingly less  
3743           stringent than the previous level. Therefore, any typeN value MAY be registered using a process for  
3744           some typeM where M is less than N, however such registration is NOT REQUIRED. For example, a  
3745           type3 value MAY be registered in a type 1 manner (by being included in a future version of an IPP  
3746           specification), however, it is NOT REQUIRED.

3747 This specification defines keyword and enum values for all of the above types, including type3 keywords.

3748 For private (unregistered) keyword extensions, implementers SHOULD use keywords with a suitable  
3749 distinguishing prefix, such as "xxx-" where xxx is the (lowercase) fully qualified company name registered  
3750 with IANA for use in domain names [RFC1035]. For example, if the company XYZ Corp. had obtained  
3751 the domain name "XYZ.com", then a private keyword 'abc' would be: 'xyz.com-abc'.

3752 Note: RFC 1035 [RFC1035] indicates that while upper and lower case letters are allowed in domain  
3753 names, no significance is attached to the case. That is, two names with the same spelling but different  
3754 case are to be treated as if identical. Also, the labels in a domain name must follow the rules for  
3755 ARPANET host names: They must start with a letter, end with a letter or digit, and have as interior  
3756 characters only letters, digits, and hyphen. Labels must be 63 characters or less. Labels are separated by  
3757 the "." character.

3758 For private (unregistered) enum extension, implementers ~~SHALL~~MUST use values in the reserved  
3759 integer range which is 2\*\*30 to 2\*\*31-1.

## 3760 6.2 Attribute Extensibility

3761 Attribute names are type2 keywords. Therefore, new attributes may be registered and have the same  
3762 status as attributes in this document by following the type2 extension rules. For private (unregistered)  
3763 attribute extensions, implementers SHOULD use keywords with a suitable distinguishing prefix as  
3764 described in Section 6.1.

3765 IANA will publish approved attribute registration specifications as separate files:

3766 ftp.isi.edu/iana/assignments/ipp/attributes/xxx-yyy.txt

3767 where "xxx-yyy" is the new attribute name.

3768 If a new Printer object attribute is defined and its values can be affected by a specific document format, its  
3769 specification needs to contain the following sentence:

3770 "The value of this attribute returned in a Get-Printer-Attributes response MAY depend on the  
3771 "document-format" attribute supplied (see Section 3.2.5.1)."

3772 If the specification does not, then its value in the Get-Printer-Attributes response ~~SHALL~~MUST NOT  
3773 depend on the "document-format" supplied in the request. When a new Job Template attribute is  
3774 registered, the value of the Printer attributes MAY vary with "document-format" supplied in the request  
3775 without the specification having to indicate so.



## 3776 6.3 Attribute Syntax Extensibility

3777 Attribute syntaxes are like type2 enums. Therefore, new attribute syntaxes may be registered and have  
3778 the same status as attribute syntaxes in this document by following the type2 extension rules described in  
3779 Section 6.1. The value codes that identify each of the attribute syntaxes are assigned in the protocol  
3780 specification [IPP-PRO], including a designated range for private, experimental use.

3781 For attribute syntaxes, the IPP Designated Expert in consultation with IANA assigns the next attribute  
3782 syntax code in the appropriate range as specified in [IPP-PRO]. IANA will publish approved attribute  
3783 syntax registration specifications as separate files:

3784 ftp.isi.edu/iana/assignments/ipp/attribute-syntaxes/xxx-yyy.txt

3785 where 'xxx-yyy' is the new attribute syntax name.

## 3786 6.4 Operation Extensibility

3787 Operations may also be registered following the type2 procedures described in Section 6.1, though major  
3788 new operations will usually be done by a new standards track RFC that augments this document. For  
3789 private (unregistered) operation extensions, implementers **SHALLMUST** use the range for the  
3790 "operation-id" in requests specified in Section 4.4.13 "operations-supported" Printer attribute.

3791 For operations, the IPP Designated Expert in consultation with IANA assigns the next operation-id code  
3792 as specified in Section 4.4.13. IANA will publish approved operation registration specifications as  
3793 separate files:

3794 ftp.isi.edu/iana/assignments/ipp/operations/Xxx-Yyy.txt

3795 where "Xxx-Yyy" is the new operation name.

## 3796 6.5 Status Code Extensibility

3797 Operation status codes may also be registered following the type2 procedures described in Section 6.1.  
3798 The values for status codes are allocated in ranges as specified in Section 14 for each status code class:

3799 "informational" - Request received, continuing process  
3800 "successful" - The action was successfully received, understood, and accepted  
3801 "redirection" - Further action must be taken in order to complete the request  
3802 "client-error" - The request contains bad syntax or cannot be fulfilled  
3803 "server-error" - The IPP object failed to fulfill an apparently valid request



3804  
3805 For private (unregistered) operation status code extensions, implementers **SHALLMUST** use the top of  
3806 each range as specified in Section 14.

3807 For operation status codes, the IPP Designated Expert in consultation with IANA assigns the next status  
3808 code in the appropriate class range as specified in Section 14. IANA will publish approved status code  
3809 registration specifications as separate files:

3810 ftp.isi.edu/iana/assignments/ipp/status-codes/xxx-yyy.txt

3811 where "xxx-yyy" is the new operation status code keyword.

## 3812 6.6 Registration of MIME types/sub-types for document-formats

3813 The "document-format" attribute's syntax is 'mimeMediaType'. This means that valid values are Internet  
3814 Media Types (see Section 4.1.9). RFC 2045 [RFC2045] defines the syntax for valid Internet media  
3815 types. IANA is the registry for all Internet media types.

## 3816 6.7 Registration of charsets for use in 'charset' attribute values

3817 The "attributes-charset" attribute's syntax is 'charset'. This means that valid values are charsets names.  
3818 When a charset in the IANA registry has more than one name (alias), the name labeled as "(preferred  
3819 MIME name)", if present, **SHALLMUST** be used (see Section 4.1.7). IANA is the registry for charsets  
3820 following the procedures of [~~RFC2278~~IANA-CSa].

## 3821 7. Internationalization Considerations

3822 Some of the attributes have values that are text strings and names which are intended for human  
3823 understanding rather than machine understanding (see the 'text' and 'name' attribute syntaxes in Sections  
3824 4.1.1 and 4.1.2).

3825 In each operation request, the client

- 3826 - identifies the charset and natural language of the request which affects each supplied 'text' and 'name'  
3827 attribute value, and
- 3828 - requests the charset and natural language for attributes returned by the IPP object in operation  
3829 responses (as described in Section 3.1.4.1).

3830

3831 In addition, the client MAY separately and individually identify the Natural Language Override of a  
3832 supplied 'text' or 'name' attribute using the 'textWithLanguage' and 'nameWithLanguage' technique  
3833 described section 4.1.1.2 and 4.1.2.2 respectively.

3834 All IPP objects **SHALLMUST** support the UTF-8 [RFC2044] charset in all 'text' and 'name' attributes  
3835 supported. If an IPP object supports more than the UTF-8 charset, the object **SHALLMUST** convert  
3836 between them in order to return the requested charset to the client according to Section 3.1.4.2. If an  
3837 IPP object supports more than one natural language, the object SHOULD return 'text' and 'name' values  
3838 in the natural language requested where those values are generated by the Printer (see Section 3.1.4.1).

3839 For Printers that support multiple charsets and/or multiple natural languages in 'text' and 'name' attributes,  
3840 different jobs may have been submitted in differing charsets and/or natural languages. All responses  
3841 **SHALLMUST** be returned in the charset requested by the client. However, the Get-Jobs operation uses  
3842 the 'textWithLanguage' and 'nameWithLanguage' mechanism to identify the differing natural languages  
3843 with each job returned.

3844 The Printer object also has configured charset and natural language attributes. The client can query the  
3845 Printer object to determine the list of charsets and natural languages supported by the Printer object and  
3846 what the Printer object's configured values are. See the "charset-configured", "charset-supported",  
3847 "natural-language-configured", and "generated-natural-language-supported" Printer description attributes  
3848 for more details.

3849 The "charset-supported" attributed identifies the supported charsets. If a charset is supported, the IPP  
3850 object MUST be capable of converting to and from that charset into any other supported charset. In  
3851 many cases, an IPP object will support only one charset and it MUST be the UTF-8 charset.

3852 The "charset-configured" attribute identifies the one supported charset which is the native charset given  
3853 the current configuration of the IPP object (administrator defined).

3854 The "generated-natural-language-supported" attribute identifies the set of supported natural languages for  
3855 generated messages; it is not related to the set of natural languages that must be accepted for client  
3856 supplied 'text' and 'name' attributes. For client supplied 'text' and 'name' attributes, an IPP object MUST  
3857 accept ALL supplied natural languages. Just because a Printer object is currently configured to support  
3858 'en-us' natural language does not mean that the Printer object should reject a job if the client supplies a  
3859 job name that is in 'fr-ca'.

3860 The "natural-language-configured" attribute identifies the one supported natural language for generated  
3861 messages which is the native natural language given the current configuration of the IPP object  
3862 (administrator defined).

3863 Attributes of type 'text' and 'name' are populated from different sources. These attributes can be  
 3864 categorized into following groups (depending on the source of the attribute):

- 3865 1. Some attributes are supplied by the client (e.g., the client supplied "job-name", "document-name",  
 3866 and "requesting-user-name" operation attributes along with the corresponding Job object's "job-  
 3867 name" and "job-originating-user-name" attributes). The IPP object MUST accept these attributes  
 3868 in any natural language no matter what the set of supported languages for generated messages  
 3869 2. Some attributes are supplied by the system administrator (e.g., the Printer object's "printer-name"  
 3870 and "printer-location" attributes). These too can be in any natural language. If the natural  
 3871 language for these attributes is different than what a client requests, then they must be reported  
 3872 using the Natural Language Override mechanism.  
 3873 3. Some attributes are supplied by the device manufacturer (e.g., the Printer object's "printer-make-  
 3874 and-model" attribute). These too can be in any natural language. If the natural language for these  
 3875 attributes is different than what a client requests, then they must be reported using the Natural  
 3876 Language Override mechanism.  
 3877 4. Some attributes are supplied by the operator (e.g., the Job object's "job-message-from-operator"  
 3878 attribute). These too can be in any natural language. If the natural language for these attributes is  
 3879 different than what a client requests, then they must be reported using the Natural Language  
 3880 Override mechanism.  
 3881 5. Some attributes are generated by the IPP object (e.g., the Job object's "job-state-message"  
 3882 attribute, the Printer object's "printer-state-message" attribute, and the "status-message" operation  
 3883 attribute). These attributes can only be in one of the "generated-natural-language-supported"  
 3884 natural languages. If a client requests some natural language for these attributes other than one of  
 3885 the supported values, the IPP object SHOULD respond in using the value of the "natural-  
 3886 language-configured" attribute (using the Natural Language Override mechanism if needed).  
 3887

3888 The 'text' and 'name' attributes specified in this version of this document (additional ones will be  
 3889 registered according to the procedures in Section 6) are:

3890	Attributes	Source
3891	-----	-----
3892	Operation Attributes	
3893	job-name (name)	client
3894	document-name (name)	client
3895	requesting-user-name (name)	client
3896		
3897	Job Attributes:	
3898	job-name (name)	client or Printer object
3899	job-originating-user-name (name)	Printer object
3900	job-state-message (text)	Job or Printer object

3901	job-message-from-operator (text)	operator
3902		
3903	Printer Attributes:	
3904	printer-name (name)	administrator
3905	printer-location (text)	administrator
3906	printer-info (text)	administrator
3907	printer-make-and-model (text)	administrator or manufacturer
3908	printer-state-message (text)	Printer object
3909	printer-message-from-operator (text)	operator

## 3910 8. Security Considerations

3911 Some IPP objects MAY be deployed over protocol stacks that support Transport Layer Security (TLS)  
 3912 Version 1.0. Other IPP objects MAY be deployed over protocol stacks that do not support TLS. Some  
 3913 IPP objects MAY be deployed over both types of protocol stacks. Those IPP objects that support TLS,  
 3914 are capable of supporting mutual authentication as well as privacy of messages via multiple encryption  
 3915 schemes. TLS 1.0 also supports a backwards compatibility mode for negotiating down to SSL3 which  
 3916 leverages the vast installed base of SSL3 aware clients and servers. An important point about security  
 3917 related information for TLS access to an IPP object, is that the security-related parameters  
 3918 (authentication, encryption keys, etc.) are "out-of-band" to the actual IPP protocol.

3919 An IPP object that does not support TLS MAY elect to support a transport layer that provides other  
 3920 security mechanisms. For example, in a mapping of IPP over HTTP/1.1 [IPP-PRO], if the IPP object  
 3921 does not support TLS, HTTP still allows for client authentication.

3922 It is difficult to anticipate the security risks that might exist in any given IPP environment. For example, if  
 3923 IPP is used within a given corporation over a private network, the risks of exposing document data may  
 3924 be low enough that the corporation will choose not to use encryption on that data. However, if the  
 3925 connection between the client and the IPP object is over a public network, the client may wish to protect  
 3926 the content of the information during transmission through the network with encryption.

3927 Furthermore, the value of the information being printed may vary from one IPP environment to the next.  
 3928 Printing payroll checks, for example, would have a different value than printing public information from a  
 3929 file. There is also the possibility of denial-of-service attacks, but denial-of-service attacks against printing  
 3930 resources are not well understood and there is no published precedents regarding this scenario.

3931 Once the authenticated identity of the requester has been supplied to the IPP object, the object uses that  
 3932 identity to enforce any authorization policy that might be in place. For example, one site's policy might  
 3933 be that only the job owner is allowed to cancel a job. The details and mechanisms to set up a particular  
 3934 access control policy are not part of IPP/1.0, and must be established via some other type of

3935 administrative or access control framework. However, there are operation status codes that allow an IPP  
3936 server to return information back to a client about any potential access control violations for an IPP  
3937 object.

3938 During a create operation, the client's identity is recorded in the Job object in an implementation-defined  
3939 attribute. This information can be used to verify a client's identity for subsequent operations on that Job  
3940 object in order to enforce any access control policy that might be in effect. See section 8.3 below for  
3941 more details.

3942 Since the security levels or the specific threats that any given IPP system administrator may be concerned  
3943 with cannot be anticipated, IPP **MUST** be capable of operating with different security mechanisms and  
3944 security policies as required by the individual installation. Security policies might vary from very strong,  
3945 to very weak, to none at all, and corresponding security mechanisms will be required. TLS Version 1.0  
3946 supports the type of negotiated levels of security required by most, if not all, potential IPP environments.  
3947 IPP environments that require no security can elect to deploy IPP objects that do not utilize the optional  
3948 TLS security mechanisms.

## 3949 8.1 Security Scenarios

3950 The following sections describe specific security attacks for IPP environments. Where examples are  
3951 provided they should be considered illustrative of the environment and not an exhaustive set. Not all of  
3952 these environments will necessarily be addressed in initial implementations of IPP.

### 3953 8.1.1 Client and Server in the Same Security Domain

3954 This environment is typical of internal networks where traditional office workers print the output of  
3955 personal productivity applications on shared work-group printers, or where batch applications print their  
3956 output on large production printers. Although the identity of the user may be trusted in this environment,  
3957 a user might want to protect the content of a document against such attacks as eavesdropping, replaying  
3958 or tampering.

### 3959 8.1.2 Client and Server in Different Security Domains

3960 Examples of this environment include printing a document created by the client on a publicly available  
3961 printer, such as at a commercial print shop; or printing a document remotely on a business associate's  
3962 printer. This latter operation is functionally equivalent to sending the document to the business associate  
3963 as a facsimile. Printing sensitive information on a Printer in a different security domain requires strong  
3964 security measures. In this environment authentication of the printer is required as well as protection  
3965 against unauthorized use of print resources. Since the document crosses security domains, protection

3966 against eavesdropping and document tampering are also required. It will also be important in this  
3967 environment to protect Printers against "spamming" and malicious document content.

### 3968 8.1.3 Print by Reference

3969 When the document is not stored on the client, printing can be done by reference. That is, the print  
3970 request can contain a reference, or pointer, to the document instead of the actual document itself.  
3971 Standard methods currently do not exist for remote entities to "assume" the credentials of a client for  
3972 forwarding requests to a 3rd party. It is anticipated that Print-By-Reference will be used to access  
3973 "public" documents and that sophisticated methods for authenticating "proxies" will not be specified for  
3974 version 1 of IPP.

## 3975 8.2 URIs for TLS and non-TLS Access

3976 As described earlier, an IPP object can support TLS access, non-TLS access, or both. The "printer-uri-  
3977 supported" attribute contains the Printer object's URI(s). Its companion attribute, "uri-security-  
3978 supported", identifies the security mechanism used for each URI listed in the "printer-uri-supported"  
3979 attribute. For each Printer operation request, a client **SHALLMUST** supply only one URI in the "printer-  
3980 uri" operation attribute. In other words, even though the Printer supports more than one URI, the client  
3981 only interacts with the Printer object using one of its URIs. This duality is not needed for Job objects,  
3982 since the Printer objects is the factory for Job objects, and the Printer object will generate the correct URI  
3983 for new Job objects depending on the Printer object's security configuration.

## 3984 8.3 The "requesting-user-name" (name(MAX)) Operation Attribute

3985 Each operation **SHALLMUST** specify the user who is performing the operation in both of the following  
3986 two ways:

- 3987 1) via the **MANDATORYREQUIRED** "requesting-user-name" operation attribute that a client  
3988 SHOULD supply in all operations. The client **SHALLMUST** obtain the value for this attribute  
3989 from an environmental or network login name for the user, rather than allowing the user to supply  
3990 any value. If the client does not supply a value for "requesting-user-name", the printer  
3991 **SHALLMUST** assume that the client is supplying some anonymous name, such as "anonymous".
- 3992 2) via an authentication mechanism of the underlying transport which may be configured to give no  
3993 authentication information.

3994  
3995 There are six cases to consider:

- 3996 a) the authentication mechanism gives no information, and the client doesn't specify "requesting-  
3997 user-name".
- 3998 b) the authentication mechanism gives no information, but the client specifies "requesting-user-  
3999 name".
- 4000 c) the authentication mechanism specifies a user which has no human readable representation, and the  
4001 client doesn't specify "requesting-user-name".
- 4002 d) the authentication mechanism specifies a user which has no human readable representation, but the  
4003 client specifies "requesting-user-name".
- 4004 e) the authentication mechanism specifies a user which has a human readable representation. The  
4005 Printer object ignores the "requesting-user-name".
- 4006 f) the authentication mechanism specifies a user who is trusted and whose name means that the value  
4007 of the "requesting-user-name", which MUST be present, is treated as the authenticated name.  
4008

4009 Note: Case "f" is intended for a tightly coupled gateway and server to work together so that the "user"  
4010 name is able to be that of the gateway client and not that of the gateway. Because most, if not all, system  
4011 vendors will initially implement IPP via a gateway into their existing print system, this mechanism is  
4012 necessary unless the authentication mechanism allows a gateway (client) to act on behalf of some other  
4013 client.

4014 The user-name has two forms:

- 4015 - one that is human readable: it is held in the ~~MANDATORYREQUIRED~~ "job-originating-user-  
4016 name" Job Description attribute which is set during the job creation operations. It is used for  
4017 presentation only, such as returning in queries or printing on start sheets
- 4018 - one for authorization: it is held in an undefined (by IPP) Job object attribute which is set by the job  
4019 creation operation. It is used to authorize other operations, such as Send-Document, Send-URI,  
4020 Cancel-Job, to determine the user when the my-jobs' attribute is specified with Get-Jobs, and to  
4021 limit what attributes and values to return with Get-Job-Attributes and Get-Jobs.  
4022

4023 The human readable user name:

- 4024 - is the value of the "requesting-user-name" for cases b, d and f.  
4025 - comes from the authentication mechanism for case e  
4026 - is some anonymous name, such as "anonymous" for cases a and c.  
4027

4028 The user name used for authorization:

- 4029 - is the value of the "requesting-user-name" for cases b and f.  
4030 - comes from the authentication mechanism for cases c, d and e  
4031 - is some anonymous name, such as "anonymous" for case a.  
4032

4033 The essence of these rules for resolving conflicting sources of user-names is that a printer implementation  
4034 is free to pick either source as long as it achieves consistent results. That is, if a user uses the same path  
4035 for a series of requests, the requests **MUST** appear to come from the same user from the standpoint of  
4036 both the human-readable user name and the user name for authorization. This rule **MUST** continue to  
4037 apply even if a request could be authenticated by two or more mechanisms. It doesn't matter which of  
4038 several authentication mechanisms a Printer uses as long as it achieves consistent results. If a client uses  
4039 more than one authentication mechanism, it is recommended that an administrator make all credentials  
4040 resolve to the same user and user-name as much as possible.

#### 4041 8.4 Restricted Queries

4042 In many IPP operations, a client supplies a list of attributes to be returned in the response. For security  
4043 reasons, an IPP object may be configured not to return all attributes (or all values) that a client requests.  
4044 The job attributes returned **MAY** depend on whether the requesting user is the same as the user that  
4045 submitted the job. The IPP object **MAY** even return none of the requested attributes. In such cases, the  
4046 status returned is the same as if the object had returned all requested attributes. The client cannot tell by  
4047 such a response whether the requested attribute was present or absent on the object.

#### 4048 8.5 IPP Security Application Profile for TLS

4049 The IPP application profile for TLS follows the standard "Mandatory Cipher Suites" requirement as  
4050 documented in the TLS specification [TLS]. Client implementations **MUST NOT** assume any other  
4051 cipher suites are supported by an IPP Printer object.

4052 If a conforming IPP object supports TLS, it **MUST** implement and support the "Mandatory Cipher  
4053 Suites" as specified in the TLS specification and **MAY** support additional cipher suites.

4054 A conforming IPP client **SHOULD** support TLS including the "Mandatory Cipher Suites" as specified in  
4055 the TLS specification. A conforming IPP client **MAY** support additional cipher suites.

4056 It is possible that due to certain government export restrictions some non-compliant versions of this  
4057 extension could be deployed. Implementations wishing to inter-operate with such non-compliant versions  
4058 **MAY** offer the TLS\_DHE\_DSS\_EXPORT\_WITH\_DES40\_CBC\_SHA mechanism. However, since 40  
4059 bit ciphers are known to be vulnerable to attack by current technology, any client which activates a 40 bit  
4060 cipher **MUST NOT** indicate to the user that the connection is completely secure from eavesdropping.



## 4061 9. References

## 4062 [ASCII]

4063 Coded Character Set - 7-bit American Standard Code for Information Interchange (ASCII),  
4064 ANSI X3.4-1986. This standard is the specification of the US-ASCII charset.

4065 ~~[CS-POL]~~

4066 ~~———— H. Alvestrand, "IETF Policy on Character Sets and Languages, work in progress <draft-~~  
4067 ~~alvestrand-charset-policy-01.txt>, August 29, 1997.~~

## 4068 [HTPP]

4069 J. Barnett, K. Carter, R. DeBry, "Initial Draft - Hypertext Printing Protocol - HTPP/1.0",  
4070 October 1996, ftp://ftp.pwg.org/pub/pwg/ipp/historic/http/overview.ps.gz

## 4071 [IANA-CON]

4072 Narte, T. and Alvestrand, H.T.: Guidelines for Writing an IANA Considerations Section in  
4073 RFCs, Work in Progress, draft-iesg-iana-considerations-04.txt, May 21, 1998.

## 4074 [IANA-CS]

4075 IANA Registry of Coded Character Sets: ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets

4076 ~~[IANA-CSa]~~

4077 ~~———— N. Freed, J. Postel: IANA CharSet Registration Procedures, Work in Progress (draft freed-~~  
4078 ~~charset-reg-02.txt).~~

## 4079 [IANA-MT]

4080 IANA Registry of Media Types: ftp://ftp.isi.edu/in-notes/iana/assignments/media-types/

4081 [IPP LPD]

4082 Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols",  
4083 draft-ietf-ipp-lpd-ipp-map-04.txt, June 1998.

4084 [IPP-PRO]

4085 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.0: Encoding and  
4086 Transport", draft-ietf-ipp-pro-06.txt, June, 1998.

4087 [IPP-RAT]

4088 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",  
4089 draft-ietf-ipp-rat-03.txt, June, 1998.

4090 [IPP-REQ]

4091 Wright, D., "Design Goals for an Internet Printing Protocol", draft-ietf-ipp-req-02.txt, June, 1998.

- 4092 ~~[IPP-PRO]~~  
4093 ~~—— Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.0: Protocol~~  
4094 ~~Specifications", draft-ietf-ipp-pro-05.txt, November, 1997.~~
- 4095 ~~[IPP-RAT]~~  
4096 ~~—— Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",~~  
4097 ~~draft-ietf-ipp-rat-02.txt, November, 1997.~~
- 4098 ~~[IPP-REQ]~~  
4099 ~~—— Wright, D., "Requirements for an Internet Printing Protocol", draft-ietf-ipp-req.txt, November,~~  
4100 ~~1997.~~
- 4101 [ISO10646-1]  
4102 ISO/IEC 10646-1:1993, "Information technology -- Universal Multiple-Octet Coded Character  
4103 Set (UCS) - Part 1: Architecture and Basic Multilingual Plane, JTC1/SC2."
- 4104 [ISO8859-1]  
4105 ISO/IEC 8859-1:1987, "Information technology -- 8-bit One-Byte Coded Character Set - Part 1:  
4106 Latin Alphabet Nr 1", 1987, JTC1/SC2.
- 4107 [ISO10175]  
4108 ISO/IEC 10175 Document Printing Application (DPA), June 1996.
- 4109 [LDPA]  
4110 T. Hastings, S. Isaacson, M. MacKay, C. Manros, D. Taylor, P. Zehler, "LDPA - Lightweight  
4111 Document Printing Application", October 1996,  
4112 ftp://ftp.pwg.org/pub/pwg/ipp/historic/ldpa/ldpa8.pdf.gz
- 4113 [P1387.4]  
4114 Kirk, M. (editor), POSIX System Administration - Part 4: Printing Interfaces, POSIX 1387.4 D8,  
4115 1994.
- 4116 [PSIS] Herriot, R. (editor), X/Open A Printing System Interoperability Specification (PSIS), August  
4117 1995.
- 4118 [PWG]  
4119 Printer Working Group, <http://www.pwg.org>.
- 4120 [RFC1035]  
4121 P. Mockapetris, "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION", RFC  
4122 1035, November 1987.

- 4123 [RFC1179]  
4124 McLaughlin, L. III, (editor), "Line Printer Daemon Protocol" RFC 1179, August 1990.
- 4125 -[RFC1630]  
4126 T. Berners-Lee, "Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression  
4127 of Names and Addresses of Objects on the Network as used in the World-Wide Web", RFC 1630,  
4128 June 1994.
- 4129 [RFC1738]  
4130 Berners-Lee, T., Masinter, L., McCahill, M. , "Uniform Resource Locators (URL)", RFC 1738,  
4131 December, 1994.
- 4132 [RFC1759]  
4133 Smith, R., Wright, F., Hastings, T., Zilles, S., and Gyllenskog, J., "Printer MIB", RFC 1759,  
4134 March 1995.
- 4135 [RFC1766]  
4136 H. Alvestrand, "-Tags for the Identification of Languages", RFC 1766, March 1995.
- 4137 [RFC1808]  
4138 R. Fielding, "Relative Uniform Resource Locators", RFC 1808, June 1995.
- 4139 [RFC1903]  
4140 J. Case, et.al., "Textual Conventions for Version 2 of the Simple Network Management Protocol  
4141 (SNMP v2)" RFC 1903, January 1996.
- 4142 [RFC1952]  
4143 P. Deutsch, "GZIP file format specification version 4.3", RFC 1952, May 1996.
- 4144 [RFC2044]  
4145 F. Yergeau, "UTF-8, a transformation format of Unicode and ISO 10646", RFC 2044, October  
4146 1996.
- 4147 [RFC2068]  
4148 R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, "Hypertext Transfer Protocol -  
4149 HTTP/1.1", RFC 2068, January 1997
- 4150 [RFC2069]  
4151 J. Franks, P. Hallam-Baker, J. Hostetler, P. Leach, A. Luotonen, E. Sink, L. Stewart, "An  
4152 Extension to HTTP: Digest Access Authentication", RFC-2069, Jan 1997.

- 4153 [RFC2045]  
4154 N. Fried, N. Borenstein, ", Multipurpose Internet Mail Extensions (MIME) Part One: Format of  
4155 Internet Message Bodies " RFC 2045, November 1996.
- 4156 [RFC2046]  
4157 Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. N. Freed & N.  
4158 Borenstein. November 1996. (Obsoletes RFC1521, RFC1522, RFC1590), RFC 2046.
- 4159 [RFC2048]  
4160 ~~N. Freed, J. Klensin & J. Postel, "Multipurpose Internet Mail Extension (MIME) Part Four:~~  
4161 ~~Registration Procedures". N. Freed, J. Klensin & J. Postel RFC 2048,--November 1996.-(Format:~~  
4162 ~~TXT=45033 bytes) (Obsoletes RFC1521, RFC1522, RFC1590) (Also BCP0013), RFC 2048.~~
- 4163 [RFC2119]  
4164 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 , March  
4165 1997
- 4166 [RFC2228]  
4167 M. Horowitz, S. Lunt, "FTP Security Extensions", RFC 2228, October 1997.
- 4168 [RFC2277]  
4169 H. Alvestrand, "IETF Policy on Character Sets and Languages" RFC 2277, January 1998.
- 4170 [RFC2278]  
4171 N. Freed, J. Postel: "IANA CharSet Registration Procedures", RFC 2278, January 1998.
- 4172 [RFC2279]  
4173 F. Yergeau , "UTF-8, a transformation format of ISO 10646", RFC 2279. January 1998.
- 4174 [RFC2316]  
4175 S. Bellovin , "Report of the IAB Security Architecture Workshop", RFC 2316, April 1998.
- 4176 [SWP]  
4177 P. Moore, B. Jahromi, S. Butler, "Simple Web Printing SWP/1.0", May 7, 1997,  
4178 ftp://ftp.pwg.org/pub/pwg/ipp/new\_PRO/swp9705.pdf

## 4179 10. Copyright Notice

4180 This document and translations of it may be copied and furnished to others, and derivative works that  
4181 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published

4182 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright  
4183 notice and this paragraph are included on all such copies and derivative works. However, this document  
4184 itself may not be modified in any way, such as by removing the copyright notice or references to the  
4185 Internet Society or other Internet organizations, except as needed for the purpose of developing Internet  
4186 standards in which case the procedures for copyrights defined in the Internet Standards process must be  
4187 followed, or as required to translate it into languages other than English.

4188 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its  
4189 successors or assigns.

4190 This document and the information contained herein is provided on an "AS IS" basis and THE  
4191 INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL  
4192 WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY  
4193 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY  
4194 RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A  
4195 PARTICULAR PURPOSE.

#### 4196 11. Author's Address

4197 Scott A. Isaacson (Editor)  
4198 Novell, Inc.  
4199 122 E 1700 S  
4200 Provo, UT 84606

4201  
4202 Phone: 801-861-7366  
4203 Fax: 801-861-2517  
4204 e-mail: sisaacson@novell.com

4205  
4206 Tom Hastings  
4207 Xerox Corporation  
4208 701 S. Aviation Blvd.  
4209 El Segundo, CA 90245

4210  
4211 Phone: 310-333-6413  
4212 Fax: 310-333-5514  
4213 e-mail: hastings@cp10.es.xerox.com

4214  
4215 Robert Herriot  
4216 Sun Microsystems Inc.

4217 901 San Antonio.Road, MPK-17  
4218 Palo Alto, CA 94303  
4219  
4220 Phone: 650-786-8995  
4221 Fax: 650-786-7077  
4222 e-mail: robert.herriot@eng.sun.com

4223  
4224 Roger deBry  
4225 HUC/003G  
4226 IBM Corporation  
4227 P.O. Box 1900  
4228 Boulder, CO 80301-9191

4229  
4230 Phone: (303) 924-4080  
4231 Fax: (303) 924-9889  
4232 e-mail: debry@vnet.ibm.com

4233  
4234 Patrick Powell  
4235 ~~San Diego State University~~ Astart Technologies  
4236 9475 Chesapeake Dr., Suite D  
4237 San Diego, CA 95123

4238  
4239 Phone: (619) 874-6543  
4240 Fax: (619) 279-8424  
4241 e-mail: papowell@~~sdsu.edu~~astart.com

4242  
4243 IPP Mailing List: [ipp@pwg.org](mailto:ipp@pwg.org)  
4244 IPP Mailing List Subscription: [ipp-request@pwg.org](mailto:ipp-request@pwg.org)  
4245 IPP Web Page: <http://www.pwg.org/ipp/>

4246  
4247 Implementers of this specification are encouraged to join IPP Mailing List in order to participate in any  
4248 discussions of clarification issues and review of registration proposals for additional attributes and values.

4249  
4250 Other Participants:

4251 Chuck Adams - Tektronix  
4252 Jeff Barnett - IBM  
4253 Ron Bergman - Dataproducts Corp.  
4254 Sylvan Butler, HP  
4255 Keith Carter, IBM Corporation  
4256 Jeff Copeland - QMS

4257 Andy Davidson - Tektronix  
4258 Mabry Dozier - QMS  
4259 Lee Farrell - Canon Information Systems  
4260 Steve Gebert - IBM  
4261 Babek Jahromi, Microsoft  
4262 David Kellerman - Northlake Software  
4263 Rick Landau - Digital  
4264 Greg LeClair - Epson  
4265 Harry Lewis - IBM  
4266 Pete Loya - HP  
4267 Ray Lutz - Cognisys  
4268 Mike MacKay, Novell, Inc.  
4269 Daniel Manchala - Underscore  
4270 Carl-Uno Manros, Xerox, ~~Corp.~~  
4271 Jay Martin - Underscore  
4272 Larry Masinter - Xerox  
4273 Stan McConnell - Xerox  
4274 Ira McDonald, High North Inc.  
4275 Paul Moore, Microsoft  
4276 Tetsuya Morita - Ricoh  
4277 Yuichi Niwa - Ricoh  
4278 Pat Nogay - IBM  
4279 Ron Norton - Printronics  
4280 Bob Pentecost - HP  
4281 Rob Rhoads - Intel  
4282 Xavier Riley - Xerox, ~~Corp.~~  
4283 David Roach - Unisys  
4284 Stuart Rowley, Kyocera  
4285 Hiroyuki Sato - Canon  
4286 Bob Setterbo - Adobe  
4287 Devon Taylor, Novell, Inc.  
4288 Mike Timperman - Lexmark  
4289 Randy Turner - Sharp  
4290 Atsushi Yuki - Kyocera  
4291 Rick Yardumian - Xerox, ~~Corp.~~  
4292 Lloyd Young - Lexmark  
4293 Bill Wagner - DPI  
4294 Jim Walker - DAZEL  
4295 Chris Wellens - Interworking Labs  
4296 Rob Whittle - Novell

4297 Don Wright - Lexmark  
4298 Peter Zehler, Xerox, ~~Corp.~~  
4299 Steve Zilles, Adobe



## 4300 12. Formats for IPP Registration Proposals

4301 In order to propose an IPP extension for registration, the proposer must submit an application to IANA  
4302 by email to "iana@iana.org" or by filling out the appropriate form on the IANA web pages  
4303 (<http://www.iana.org>). This section specifies the required information and the formats for proposing  
4304 registrations of extensions to IPP as provided in Section 6 for:

- 4305
- 4306 1. type2 'keyword' attribute values
  - 4307 2. type3 'keyword' attribute values
  - 4308 3. type2 'enum' attribute values
  - 4309 4. type3 'enum' attribute values
  - 4310 5. attributes
  - 4311 6. attribute syntaxes
  - 4312 7. operations
  - 4313 8. status codes

## 4314 12.1 Type2 keyword attribute values registration

4315 Type of registration: type2 keyword attribute value  
4316 Name of attribute to which this keyword specification is to be added:  
4317 Proposed keyword name of this keyword value:  
4318 Specification of this keyword value (follow the style of IPP Model Section 4.1.3):  
4319 Name of proposer:  
4320 Address of proposer:  
4321 Email address of proposer:  
4322

4323 Note: For type2 keywords, the Designated Expert will be the point of contact for the approved  
4324 registration specification, if any maintenance of the registration specification is needed.

## 4325 12.2 Type3 keyword attribute values registration

4326 Type of registration: type3 keyword attribute value  
4327 Name of attribute to which this keyword specification is to be added:  
4328 Proposed keyword name of this keyword value:  
4329 Specification of this keyword value (follow the style of IPP Model Section 4.1.3):  
4330 Name of proposer:  
4331 Address of proposer:  
4332 Email address of proposer:  
4333

4334 Note: For type3 keywords, the proposer will be the point of contact for the approved registration  
4335 specification, if any maintenance of the registration specification is needed.

### 4336 12.3 Type2 enum attribute values registration

4337 Type of registration: type2 enum attribute value  
4338 Name of attribute to which this enum specification is to be added:  
4339 Keyword symbolic name of this enum value:  
4340 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):  
4341 Specification of this enum value (follow the style of IPP Model Section 4.1.4):  
4342 Name of proposer:  
4343 Address of proposer:  
4344 Email address of proposer:  
4345

4346 Note: For type2 enums, the Designated Expert will be the point of contact for the approved registration  
4347 specification, if any maintenance of the registration specification is needed.

### 4348 12.4 Type3 enum attribute values registration

4349 Type of registration: type3 enum attribute value  
4350 Name of attribute to which this enum specification is to be added:  
4351 Keyword symbolic name of this enum value:  
4352 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):  
4353 Specification of this enum value (follow the style of IPP Model Section 4.1.4):  
4354 Name of proposer:  
4355 Address of proposer:  
4356 Email address of proposer:  
4357

4358 Note: For type3 enums, the proposer will be the point of contact for the approved registration  
4359 specification, if any maintenance of the registration specification is needed.

### 4360 12.5 Attribute registration

4361 Type of registration: attribute  
4362 Proposed keyword name of this attribute:  
4363 Types of attribute (Operation, Job Template, Job Description, Printer Description):  
4364 Operations to be used with if the attribute is an operation attribute:  
4365 Object (Job, Printer, etc. if bound to an object):  
4366 Attribute syntax(es) (include 1setOf and range as in Section 4.2):  
4367 If attribute syntax is 'keyword' or 'enum', is it type2 or type3:  
4368 If this is a Printer attribute, MAY the value returned depend on "document-format" (See Section 6.2):

4369 Specification of this attribute (follow the style of IPP Model Section 4.2):

4370 Name of proposer:

4371 Address of proposer:

4372 Email address of proposer:

4373

4374 Note: For attributes, the IPP Designated Expert will be the point of contact for the approved registration  
4375 specification, if any maintenance of the registration specification is needed.

## 4376 12.6 Attribute Syntax registration

4377 Type of registration: attribute syntax

4378 Proposed name of this attribute syntax:

4379 Type of attribute syntax (integer, octetString, character-string, see [IPP-PRO]):

4380 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

4381 Specification of this attribute (follow the style of IPP Model Section 4.1):

4382 Name of proposer:

4383 Address of proposer:

4384 Email address of proposer:

4385

4386 Note: For attribute syntaxes, the IPP Designated Expert will be the point of contact for the approved  
4387 registration specification, if any maintenance of the registration specification is needed.

## 4388 12.7 Operation registration

4389 Type of registration: operation

4390 Proposed name of this operation:

4391 Numeric operation-id value (to be assigned by the IPP Designated Expert in consultation with IANA):

4392 Object Target (Job, Printer, etc. that operation is upon):

4393 Specification of this attribute (follow the style of IPP Model Section 3):

4394 Name of proposer:

4395 Address of proposer:

4396 Email address of proposer:

4397

4398 Note: For operations, the IPP Designated Expert will be the point of contact for the approved  
4399 registration specification, if any maintenance of the registration specification is needed.

## 4400 12.8 Status code registration

4401 Type of registration: status code

4402 Keyword symbolic name of this status code value:

4403 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

4404 Operations that this status code may be used with:  
4405 Specification of this status code (follow the style of IPP Model Section 14 APPENDIX B: Status Codes  
4406 and Suggested Status Code Messages):  
4407 Name of proposer:  
4408 Address of proposer:  
4409 Email address of proposer:  
4410  
4411 Note: For status codes, the Designated Expert will be the point of contact for the approved registration  
4412 specification, if any maintenance of the registration specification is needed.

### 4413 13. APPENDIX A: Terminology

4414 This specification uses the terminology defined in this section.

#### 4415 13.1 Conformance Terminology

4416 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",  
4417 "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be  
4418 interpreted as described in RFC 2119 [RFC2119]. ~~The sections below reiterate these definitions and~~  
4419 ~~include some additional ones.~~

##### 4420 ~~13.1.1 MUST~~

4421 ~~This word, or the terms "REQUIRED", "SHALL" or "MANDATORY", means that the definition is an~~  
4422 ~~absolute requirement of the specification.~~

##### 4423 ~~13.1.2 MUST NOT~~

4424 ~~This phrase, or the phrase "SHALL NOT", means that the definition is an absolute prohibition of the~~  
4425 ~~specification.~~

##### 4426 ~~13.1.3 SHOULD~~

4427 ~~This word, or the adjective "RECOMMENDED", means that there may exist valid reasons in particular~~  
4428 ~~circumstances to ignore a particular item, but the full implications must be understood and carefully~~  
4429 ~~weighed before choosing a different course.~~

~~4430 13.1.4 SHOULD NOT~~

~~4431 This phrase, or the phrase "NOT RECOMMENDED" means that there may exist valid reasons in  
4432 particular circumstances when the particular behavior is acceptable or even useful, but the full  
4433 implications should be understood and the case carefully weighed before implementing any behavior  
4434 described with this label.~~

~~4435 13.1.5 MAY~~

~~4436 This word, or the adjective "OPTIONAL", means that an item is truly optional. One vendor may choose  
4437 to include the item because a particular marketplace requires it or because the vendor feels that it  
4438 enhances the product while another vendor may omit the same item. An implementation which does not  
4439 include a particular option MUST be prepared to inter-operate with another implementation which does  
4440 include the option, though perhaps with reduced functionality. In the same vein an implementation which  
4441 does include a particular option MUST be prepared to inter-operate with another implementation which  
4442 does not include the option (except, of course, for the feature the option provides.)~~

~~4443 13.1.6~~13.1.1 NEED NOT

~~4444 This term is not included in RFC 2119.~~ The verb "NEED NOT" indicates an action that the subject of  
4445 the sentence does not have to implement in order to claim conformance to the standard. The verb  
4446 "NEED NOT" is used instead of "MAY NOT" since "MAY NOT" sounds like a prohibition.

## 4447 13.2 Model Terminology

## 4448 13.2.1 Keyword

4449 Keywords are used within this document as identifiers of semantic entities within the abstract model (see  
4450 section 4.1.3). Attribute names, some attribute values, attribute syntaxes, and attribute group names are  
4451 represented as keywords.

## 4452 13.2.2 Attributes

4453 An attribute is an item of information that is associated with an instance of an IPP object. An attribute  
4454 consists of an attribute name and one or more attribute values. Each attribute has a specific attribute  
4455 syntax. All object attributes are defined in section 4 and all operation attributes are defined in section 3.

4456 Job Template Attributes are described in section 4.2. The client optionally supplies Job Template  
4457 attributes in a create request (operation requests that create Job objects). The Printer object has  
4458 associated attributes which define supported and default values for the Printer.

## 4459 13.2.2.1 Attribute Name

4460 Each attribute is uniquely identified in this document by its attribute name. An attribute name is a  
4461 keyword. The keyword attribute name is given in the section header describing that attribute. In running  
4462 text in this document, attribute names are indicated inside double quotation marks (") where the  
4463 quotation marks are not part of the keyword itself.

## 4464 13.2.2.2 Attribute Group Name

4465 Related attributes are grouped into named groups. The name of the group is a keyword. The group  
4466 name may be used in place of naming all the attributes in the group explicitly. Attribute groups are  
4467 defined in section 3.

## 4468 13.2.2.3 Attribute Value

4469 Each attribute has one or more values. Attribute values are represented in the syntax type specified for  
4470 that attribute. In running text in this document, attribute values are indicated inside single quotation  
4471 marks ('), whether their attribute syntax is keyword, integer, text, etc. where the quotation marks are not  
4472 part of the value itself.

## 4473 13.2.2.4 Attribute Syntax

4474 Each attribute is defined using an explicit syntax type. In this document, each syntax type is defined as a  
4475 keyword with specific meaning. The protocol specification document [IPP-PRO] indicates the actual  
4476 "on-the-wire" encoding rules for each syntax type. Attribute syntax types are defined in section 4.1.

## 4477 13.2.3 Supports

4478 By definition, a Printer object supports an attribute only if that Printer object responds with the  
4479 corresponding attribute populated with some value(s) in a response to a query for that attribute. A  
4480 Printer object supports an attribute value if the value is one of the Printer object's "supported values"  
4481 attributes. The device behind a Printer object may exhibit a behavior that corresponds to some IPP  
4482 attribute, but if the Printer object, when queried for that attribute, doesn't respond with the attribute, then  
4483 as far as IPP is concerned, that implementation does not support that feature. If the Printer object's "xxx-  
4484 supported" attribute is not populated with a particular value (even if that value is a legal value for that  
4485 attribute), then that Printer object does not support that particular value.

4486 A conforming implementation ~~SHALL~~**MUST** support all ~~MANDATORY~~**REQUIRED** attributes.  
4487 However, even for ~~MANDATORY~~**REQUIRED** attributes, conformance to IPP does not mandate that all  
4488 implementations support all possible values representing all possible job processing behaviors and

4489 features. For example, if a given instance of a Printer supports only certain document formats, then that  
4490 Printer responds with the "document-format-supported" attribute populated with a set of values, possibly  
4491 only one, taken from the entire set of possible values defined for that attribute. This limited set of values  
4492 represents the Printer's set of supported document formats. Supporting an attribute and some set of  
4493 values for that attribute enables IPP end users to be aware of and make use of those features associated  
4494 with that attribute and those values. If an implementation chooses to not support an attribute or some  
4495 specific value, then IPP end users would have no ability to make use of that feature within the context of  
4496 IPP itself. However, due to existing practice and legacy systems which are not IPP aware, there might be  
4497 some other mechanism outside the scope of IPP to control or request the "unsupported" feature (such as  
4498 embedded instructions within the document data itself).

4499 For example, consider the "finishings-supported" attribute.

- 4500 1) If a Printer object is not physically capable of stapling, the "finishings-supported" attribute MUST  
4501 NOT be populated with the value of 'staple'.
- 4502 2) A Printer object is physically capable of stapling, however an implementation chooses not to  
4503 support stapling in the IPP "finishings" attribute. In this case, 'staple' ~~SHALL~~**MUST** NOT be a  
4504 value in the "finishings-supported" Printer object attribute. Without support for the value 'staple',  
4505 an IPP end user would have no means within the protocol itself to request that a Job be stapled.  
4506 However, an existing document data formatter might be able to request that the document be  
4507 stapled directly with an embedded instruction within the document data. In this case, the IPP  
4508 implementation does not "support" stapling, however the end user is still able to have some  
4509 control over the stapling of the completed job.
- 4510 3) A Printer object is physically capable of stapling, and an implementation chooses to support  
4511 stapling in the IPP "finishings" attribute. In this case, 'staple' ~~SHALL~~**MUST** be a value in the  
4512 "finishings-supported" Printer object attribute. Doing so, would enable end users to be aware of  
4513 and make use of the stapling feature using IPP attributes.

4514  
4515 Even though support for Job Template attributes by a Printer object is OPTIONAL, it is  
4516 RECOMMENDED that if the device behind a Printer object is capable of realizing any feature or  
4517 function that corresponds to an IPP attribute and some associated value, then that implementation  
4518 SHOULD support that IPP attribute and value.

4519 The set of values in any of the supported value attributes is set (populated) by some administrative  
4520 process or automatic sensing mechanism that is outside the scope of IPP. For administrative policy and  
4521 control reasons, an administrator may choose to make only a subset of possible values visible to the end  
4522 user. In this case, the real output device behind the IPP Printer abstraction may be capable of a certain  
4523 feature, however an administrator is specifying that access to that feature not be exposed to the end user  
4524 through the IPP protocol. Also, since a Printer object may represent a logical print device (not just a  
4525 physical device) the actual process for supporting a value is undefined and left up to the implementation.

4526 However, if a Printer object supports a value, some manual human action may be needed to realize the  
4527 semantic action associated with the value, but no end user action is required.

4528 For example, if one of the values in the "finishings-supported" attribute is 'staple', the actual process  
4529 might be an automatic staple action by a physical device controlled by some command sent to the device.  
4530 Or, the actual process of stapling might be a manual action by an operator at an operator attended Printer  
4531 object.

4532 For another example of how supported attributes function, consider a system administrator who desires  
4533 to control all print jobs so that no job sheets are printed in order to conserve paper. To force no job  
4534 sheets, the system administrator sets the only supported value for the "job-sheets-supported" attribute to  
4535 'none'. In this case, if a client requests anything except 'none', the create request is rejected or the "job-  
4536 sheets" value is ignored (depending on the value of "ipp-attribute-fidelity"). To force the use of job  
4537 start/end sheets on all jobs, the administrator does not include the value 'none' in the "job-sheets-  
4538 supported" attribute. In this case, if a client requests 'none', the create request is rejected or the "job-  
4539 sheets" value is ignored (again depending on the value of "ipp-attribute-fidelity").

#### 4540 13.2.4 print-stream page

4541 A "print-stream page" is a page according to the definition of pages in the language used to express the  
4542 document data.

#### 4543 13.2.5 impression

4544 An "impression" is the image (possibly many print-stream pages in different configurations) imposed onto  
4545 a single media page.

### 4546 14. APPENDIX B: Status Codes and Suggested Status Code Messages

4547 This section defines status code enum keywords and values that are used to provide semantic information  
4548 on the results of an operation request. Each operation response **MUST** include a status code. The  
4549 response **MAY** also contain a status message that provides a short textual description of the status. The  
4550 status code is intended for use by automata, and the status message is intended for the human end user.  
4551 Since the status message is an **OPTIONAL** component of the operation response, an IPP application (i.e.,  
4552 a browser, GUI, print driver or gateway) is **NOT REQUIRED** to examine or display the status message,  
4553 since it **MAY** not be returned to the application.

4554 The prefix of the status keyword defines the class of response as follows:



4555 "informational" - Request received, continuing process  
4556 "successful" - The action was successfully received, understood, and accepted  
4557 "redirection" - Further action must be taken in order to complete the request  
4558 "client-error" - The request contains bad syntax or cannot be fulfilled  
4559 "server-error" - The IPP object failed to fulfill an apparently valid request

4560

4561 As with type2 enums, IPP status codes are extensible. IPP clients are NOT REQUIRED to understand  
4562 the meaning of all registered status codes, though such understanding is obviously desirable. However,  
4563 IPP clients **SHALLMUST** understand the class of any status code, as indicated by the prefix, and treat  
4564 any unrecognized response as being equivalent to the first status code of that class, with the exception  
4565 that an unrecognized response **SHALLMUST** NOT be cached. For example, if an unrecognized status  
4566 code of "client-error-xxx-yyy" is received by the client, it can safely assume that there was something  
4567 wrong with its request and treat the response as if it had received a "client-error-bad-request" status  
4568 code. In such cases, IPP applications SHOULD present the OPTIONAL message (if present) to the end  
4569 user since the message is likely to contain human readable information which will help to explain the  
4570 unusual status. The name of the enum is the suggested status message for US English.

4571 The status code values range from 0x0000 to 0x7FFF. The value ranges for each status code class are as  
4572 follows:

4573 "successful" - 0x0000 to 0x00FF  
4574 "informational" - 0x0100 to 0x01FF  
4575 "redirection" - 0x0200 to 0x02FF  
4576 "client-error" - 0x0400 to 0x04FF  
4577 "server-error" - 0x0500 to 0x05FF

4578

4579 The top half (128 values) of each range (0x0n40 to 0x0nFF, for n = 0 to 5) is reserved for private use  
4580 within each status code class. Values 0x0600 to 0x7FFF are reserved for future assignment and  
4581 **SHALLMUST** NOT be used.

## 4582 14.1 Status Codes

4583 Each status code is described below. Section 14.2 contains a table that indicates which status codes apply  
4584 to which operations. Sections 16.3 and 16.4 describe the suggested steps for processing IPP attributes  
4585 for all operations, including returning status codes.

### 4586 14.1.1 Informational

4587 This class of status code indicates a provisional response and is to be used for informational purposes  
4588 only.

4589 There are no status codes defined in IPP/1.0 for this class of status code.

#### 4590 14.1.2 Successful Status Codes

4591 This class of status code indicates that the client's request was successfully received, understood, and  
4592 accepted.

##### 4593 14.1.2.1 successful-ok (0x0000)

4594 The request has succeeded. In the case of a response to a create request, the 'successful-ok' status code  
4595 indicates that the request was successfully received and validated, and that the Job object has been  
4596 created; it does not indicate that the job has been processed. The transition of the Job object into the  
4597 'completed' state is the only indicator that the job has been printed.

##### 4598 14.1.2.2 successful-ok-ignored-or-substituted-attributes (0x0001)

4599 The request has succeeded, but some attributes were ignored or unsupported values were substituted  
4600 with supported values in order to process the job without rejecting it.

##### 4601 14.1.2.3 successful-ok-conflicting-attributes (0x0002)

4602 The request has succeeded, but some attribute values conflicted with the values of other attributes. These  
4603 conflicting values were either (1) substituted with (supported) values or (2) the attributes were removed  
4604 in order to process the job without rejecting it.

#### 4605 14.1.3 Redirection Status Codes

4606 This class of status code indicates that further action needs to be taken to fulfill the request.

4607 There are no status codes defined in IPP/1.0 for this class of status code.

#### 4608 14.1.4 Client Error Status Codes

4609 This class of status code is intended for cases in which the client seems to have erred. The IPP object  
4610 SHOULD return a message containing an explanation of the error situation and whether it is a temporary  
4611 or permanent condition.

## 4612 14.1.4.1 client-error-bad-request (0x0400)

4613 The request could not be understood by the IPP object due to malformed syntax (such as the value of a  
4614 fixed length attribute whose length does not match the prescribed length for that attribute - see section  
4615 16.3). The IPP application SHOULD NOT repeat the request without modifications.

## 4616 14.1.4.2 client-error-forbidden (0x0401)

4617 The IPP object understood the request, but is refusing to fulfill it. Additional authentication information  
4618 or authorization credentials will not help and the request SHOULD NOT be repeated. This status code is  
4619 commonly used when the IPP object does not wish to reveal exactly why the request has been refused or  
4620 when no other response is applicable.

## 4621 14.1.4.3 client-error-not-authenticated (0x0402)

4622 The request requires user authentication. The IPP client may repeat the request with suitable  
4623 authentication information. If the request already included authentication information, then this status  
4624 code indicates that authorization has been refused for those credentials. If this response contains the  
4625 same challenge as the prior response, and the user agent has already attempted authentication at least  
4626 once, then the response message may contain relevant diagnostic information. This status codes reveals  
4627 more information than "client-error-forbidden".

## 4628 14.1.4.4 client-error-not-authorized (0x0403)

4629 The requester is not authorized to perform the request. Additional authentication information or  
4630 authorization credentials will not help and the request SHOULD NOT be repeated. This status code is  
4631 used when the IPP object wishes to reveal that the authentication information is understandable, however,  
4632 the requester is explicitly not authorized to perform the request. This status codes reveals more  
4633 information than "client-error-forbidden" and "client-error-not-authenticated".

## 4634 14.1.4.5 client-error-not-possible (0x0404)

4635 This status code is used when the request is for something that can not happen. For example, there might  
4636 be a request to cancel a job that has already been canceled or aborted by the system. The IPP client  
4637 SHOULD NOT repeat the request.

## 4638 14.1.4.6 client-error-timeout (0x0405)

4639 The client did not produce a request within the time that the IPP object was prepared to wait. For  
4640 example, a client issued a Create-Job operation and then, after a long period of time, issued a Send-

4641 Document operation and this error status code was returned in response to the Send-Document request  
4642 (see section 3.3.1). The IPP object might have been forced to clean up resources that had been held for  
4643 the waiting additional Documents. The IPP object was forced to close the Job since the client took too  
4644 long. The client SHOULD NOT repeat the request without modifications.

#### 4645 14.1.4.7 client-error-not-found (0x0406)

4646 The IPP object has not found anything matching the request URI. No indication is given of whether the  
4647 condition is temporary or permanent. For example, a client with an old reference to a Job (a URI) tries to  
4648 cancel the Job, however in the mean time the Job might have been completed and all record of it at the  
4649 Printer has been deleted. This status code, 'client-error-not-found' is returned indicating that the  
4650 referenced Job can not be found. This error status code is also used when a client supplies a URI as a  
4651 reference to the document data in either a Print-URI or Send-URI operation, but the document can not  
4652 be found.

4653 In practice, an IPP application should avoid a not found situation by first querying and presenting a list of  
4654 valid Printer URIs and Job URIs to the end-user.

#### 4655 14.1.4.8 client-error-gone (0x0407)

4656 The requested object is no longer available and no forwarding address is known. This condition should  
4657 be considered permanent. Clients with link editing capabilities should delete references to the request  
4658 URI after user approval. If the IPP object does not know or has no facility to determine, whether or not  
4659 the condition is permanent, the status code "client-error-not-found" should be used instead.

4660 This response is primarily intended to assist the task of maintenance by notifying the recipient that the  
4661 resource is intentionally unavailable and that the IPP object administrator desires that remote links to that  
4662 resource be removed. It is not necessary to mark all permanently unavailable resources as "gone" or to  
4663 keep the mark for any length of time -- that is left to the discretion of the IPP object administrator.

#### 4664 14.1.4.9 client-error-request-entity-too-large (0x0408)

4665 The IPP object is refusing to process a request because the request entity is larger than the IPP object is  
4666 willing or able to process. An IPP Printer returns this status code when it limits the size of print jobs and  
4667 it receives a print job that exceeds that limit or when the attributes are so many that their encoding causes  
4668 the request entity to exceed IPP object capacity.

## 4669 14.1.4.10 client-error-request-value-too-long (0x0409)

4670 The IPP object is refusing to service the request because one or more of the ~~client~~-client-supplied  
4671 attributes has a variable length value that is longer than the maximum length specified for that attribute.  
4672 The IPP object might not have sufficient resources (memory, buffers, etc.) to process (even temporarily),  
4673 interpret, and/or ignore ~~the large a~~ value larger than the maximum length. Another use of this error  
4674 code is when the IPP object supports the processing of ~~a the~~-large value that is less than the maximum  
4675 length, but during the processing of the request as a whole, the object may pass the value onto some  
4676 other system component which is not able to accept the large value. For more details, see section 16.3.

4677 Note: For attribute values that are URIs, this rare condition is only likely to occur when a client has  
4678 improperly submitted a request with long query information (e.g. an IPP application allows an end-user to  
4679 enter an invalid URI), when the client has descended into a URI "black hole" of redirection (e.g., a  
4680 redirected URI prefix that points to a suffix of itself), or when the IPP object is under attack by a client  
4681 attempting to exploit security holes present in some IPP objects using fixed-length buffers for reading or  
4682 manipulating the Request-URI.

## 4683 14.1.4.11 client-error-document-format-not-supported (0x040A)

4684 The IPP object is refusing to service the request because the document data is in a format, as specified in  
4685 the "document-format" operation attribute, that is not supported by the Printer object. This error is  
4686 returned independent of the client-supplied "ipp-attribute-fidelity". The Printer object **SHALLMUST**  
4687 return this status code, even if there are other attributes that are not supported as well, since this error is  
4688 a bigger problem than with Job Template attributes.

## 4689 14.1.4.12 client-error-attributes-or-values-not-supported (0x040B)

4690 In a create request, if the Printer object does not support one or more attributes or attribute values  
4691 supplied in the request and the client supplied the "ipp-attributes-fidelity" operation attribute with the  
4692 'true' value, the Printer object **shallMUST** return this status code. For example, if the request indicates  
4693 'iso-a4' media, but that media type is not supported by the Printer object. Or, if the client supplies an  
4694 optional attribute and the attribute itself is not even supported by the Printer. If the "ipp-attribute-  
4695 fidelity" attribute is 'false', the Printer **SHALLMUST** ignore or substitute values for unsupported  
4696 attributes and values rather than reject the request and return this status code.

4697 For any operation where a client requests attributes (such as a Get-Jobs, Get-Printer-Attributes, or Get-  
4698 Job-Attributes operation), if the IPP object does not support one or more of the requested attributes, the  
4699 IPP object simply ignores the unsupported requested attributes and processes the request as if they had  
4700 not been supplied, rather than returning this status code.

## 4701 14.1.4.13 client-error-uri-scheme-not-supported (0x040C)

4702 The type of the client supplied URI in a Print-URI or a Send-URI operation is not supported.

## 4703 14.1.4.14 client-error-charset-not-supported (0x040D)

4704 For any operation, if the IPP Printer does not support the charset supplied by the client in the "attributes-  
4705 charset" operation attribute, the Printer **SHALLMUST** reject the operation and return this status (see  
4706 Section 3.1.4.1).

## 4707 14.1.4.15 client-error-conflicting-attributes (0x040E)

4708 The request is rejected because some attribute values conflicted with the values of other attributes.

## 4709 14.1.5 Server Error Status Codes

4710 This class of status codes indicates cases in which the IPP object is aware that it has erred or is incapable  
4711 of performing the request. The IPP object SHOULD include a message containing an explanation of the  
4712 error situation, and whether it is a temporary or permanent condition.

## 4713 14.1.5.1 server-error-internal-error (0x0500)

4714 The IPP object encountered an unexpected condition that prevented it from fulfilling the request. This  
4715 error status code differs from "server-error-temporary-error" in that it implies a more permanent type of  
4716 internal error. It also differs from "server-error-device-error" in that it implies an unexpected condition  
4717 (unlike a paper-jam or out-of-toner problem which is undesirable but expected). This error status code  
4718 indicates that probably some knowledgeable human intervention is required.

## 4719 14.1.5.2 server-error-operation-not-supported (0x0501)

4720 The IPP object does not support the functionality required to fulfill the request. This is the appropriate  
4721 response when the IPP object does not recognize an operation or is not capable of supporting it.

## 4722 14.1.5.3 server-error-service-unavailable (0x0502)

4723 The IPP object is currently unable to handle the request due to a temporary overloading or maintenance  
4724 of the IPP object. The implication is that this is a temporary condition which will be alleviated after some  
4725 delay. If known, the length of the delay may be indicated in the message. If no delay is given, the IPP  
4726 application should handle the response as it would for a "server-error-temporary-error" response. If the

4727 condition is more permanent, the error status codes "client-error-gone" or "client-error-not-found" could  
4728 be used.

#### 4729 14.1.5.4 server-error-version-not-supported (0x0503)

4730 The IPP object does not support, or refuses to support, the IPP protocol version that was used in the  
4731 request message. The IPP object is indicating that it is unable or unwilling to complete the request using  
4732 the same version as supplied in the request other than with this error message. The response should  
4733 contain a Message describing why that version is not supported and what other versions are supported by  
4734 that IPP object.

4735 A conforming IPP/1.0 client **SHALLMUST** specify the valid version ('1.0') on each request. A  
4736 conforming IPP/1.0 object **SHALLMUST** NOT return this status code to a conforming IPP/1.0 client.  
4737 An IPP object **SHALLMUST** return this status code to a non-conforming IPP client. The response  
4738 **SHALLMUST** identify in the "version-number" operation attribute the closest version number that the  
4739 IPP object does support.

#### 4740 14.1.5.5 server-error-device-error (0x0504)

4741 A printer error, such as a paper jam, occurs while the IPP object processes a Print or Send operation.  
4742 The response contains the true Job Status (the values of the "job-state" and "job-state-reasons"  
4743 attributes). Additional information can be returned in the optional "job-state-message" attribute value or  
4744 in the OPTIONAL status message that describes the error in more detail. This error status code is only  
4745 returned in situations where the Printer is unable to accept the create request because of such a device  
4746 error. For example, if the Printer is unable to spool, and can only accept one job at a time, the reason it  
4747 might reject a create request is that the printer currently has a paper jam. In many cases however, where  
4748 the Printer object can accept the request even though the Printer has some error condition, the  
4749 'successful-ok' status code will be returned. In such a case, the client would look at the returned Job  
4750 Object Attributes or later query the Printer to determine its state and state reasons.

#### 4751 14.1.5.6 server-error-temporary-error (0x0505)

4752 A temporary error such as a buffer full write error, a memory overflow (i.e. the document data exceeds  
4753 the memory of the Printer), or a disk full condition, occurs while the IPP Printer processes an operation.  
4754 The client MAY try the unmodified request again at some later point in time with an expectation that the  
4755 temporary internal error condition may have been cleared. Alternatively, as an implementation option, a  
4756 Printer object MAY delay the response until the temporary condition is cleared so that no error is  
4757 returned.

## 4758 14.1.5.7 server-error-not-accepting-jobs (0x0506)

4759 A temporary error indicating that the Printer is not currently accepting jobs, because the administrator has  
4760 set the value of the Printer's "printer-is-not-accepting-jobs" attribute to 'false' (by means outside of  
4761 IPP/1.0).

## 4762 14.1.5.8 server-error-busy (0x0507)

4763 A temporary error indicating that the Printer is too busy processing jobs and/or other requests. The client  
4764 SHOULD try the unmodified request again at some later point in time with an expectation that the  
4765 temporary busy condition will have been cleared.



## 4766 14.2 Status Codes for IPP Operations

4767 PJ = Print-Job, PU = Print-URI, CJ = Create-Job, SD = Send-Document  
 4768 SU = Send-URI, V = Validate-Job, GA = Get-Job-Attributes and  
 4769 Get-Printer-Attributes, GJ = Get-Jobs, C = Cancel-Job

4770		IPP Operations								
4771	IPP Status Keyword	PJ	PU	CJ	SD	SU	V	GA	GJ	C
4772	-----	--	--	--	--	--	--	--	--	--
4773	successful-ok	x	x	x	x	x	x	x	x	x
4774	successful-ok-ignored-or-substituted-	x	x	x	x	x	x	x	x	x
4775	attributes									
4776	successful-ok-conflicting-attributes	x	x	x	x	x	x	x	x	x
4777	client-error-bad-request	x	x	x	x	x	x	x	x	x
4778	client-error-forbidden	x	x	x	x	x	x	x	x	x
4779	client-error-not-authenticated	x	x	x	x	x	x	x	x	x
4780	client-error-not-authorized	x	x	x	x	x	x	x	x	x
4781	client-error-not-possible	x	x	x	x	x	x	x	x	x
4782	client-error-timeout	x	x	x	x	x	x	x	x	x
4783	client-error-not-found	x	x	x	x	x	x	x	x	x
4784	client-error-gone	x	x	x	x	x	x	x	x	x
4785	client-error-request-entity-too-large	x	x	x	x	x	x	x	x	x
4786	client-error-request-value-too-long	x	x	x	x	x	x	x	x	x
4787	client-error-document-format-not-	x	x		x	x	x	x		
4788	supported									
4789	client-error-attributes-or-values-not-	x	x	x	x	x	x	x	x	x
4790	supported									
4791	client-error-uri-scheme-not-supported		x			x				
4792	client-error-charset-not-supported	x	x	x	x	x	x	x	x	x
4793	client-error-conflicting-attributes	x	x	x	x	x	x	x	x	x
4794	server-error-internal-error	x	x	x	x	x	x	x	x	x
4795	server-error-operation-not-supported		x	x	x	x				
4796	server-error-service-unavailable	x	x	x	x	x	x	x	x	x
4797	server-error-version-not-supported	x	x	x	x	x	x	x	x	x
4798	server-error-device-error	x	x	x	x	x				
4799	server-error-temporary-error	x	x	x	x	x				
4800	server-error-not-accepting-jobs	x	x	x	x	x	x			
4801	server-error-busy	x	x	x	x	x	x	x	x	x
4802										
4803										
4804										

## 4805 15. APPENDIX C: "media" keyword values

4806 Standard keyword values are taken from several sources.

4807 Standard values are defined (taken from DPA[ISO10175] and the Printer MIB[RFC1759]):

4808 'default': The default medium for the output device  
4809 'iso-a4-white': Specifies the ISO A4 white medium  
4810 'iso-a4-colored': Specifies the ISO A4 colored medium  
4811 'iso-a4-transparent' Specifies the ISO A4 transparent medium  
4812 'iso-a3-white': Specifies the ISO A3 white medium  
4813 'iso-a3-colored': Specifies the ISO A3 colored medium  
4814 'iso-a5-white': Specifies the ISO A5 white medium  
4815 'iso-a5-colored': Specifies the ISO A5 colored medium  
4816 'iso-b4-white': Specifies the ISO B4 white medium  
4817 'iso-b4-colored': Specifies the ISO B4 colored medium  
4818 'iso-b5-white': Specifies the ISO B5 white medium  
4819 'iso-b5-colored': Specifies the ISO B5 colored medium  
4820 'jis-b4-white': Specifies the JIS B4 white medium  
4821 'jis-b4-colored': Specifies the JIS B4 colored medium  
4822 'jis-b5-white': Specifies the JIS B5 white medium  
4823 'jis-b5-colored': Specifies the JIS B5 colored medium  
4824

4825 The following standard values are defined for North American media:

4826 'na-letter-white': Specifies the North American letter white medium  
4827 'na-letter-colored': Specifies the North American letter colored medium  
4828 'na-letter-transparent': Specifies the North American letter transparent medium  
4829 'na-legal-white': Specifies the North American legal white medium  
4830 'na-legal-colored': Specifies the North American legal colored medium  
4831

4832 The following standard values are defined for envelopes:

4833 'iso-b4-envelope': Specifies the ISO B4 envelope medium  
4834 'iso-b5-envelope': Specifies the ISO B5 envelope medium  
4835 'iso-c3-envelope': Specifies the ISO C3 envelope medium  
4836 'iso-c4-envelope': Specifies the ISO C4 envelope medium  
4837 'iso-c5-envelope': Specifies the ISO C5 envelope medium  
4838 'iso-c6-envelope': Specifies the ISO C6 envelope medium  
4839 'iso-designated-long-envelope': Specifies the ISO Designated Long envelope medium  
4840 'na-10x13-envelope': Specifies the North American 10x13 envelope medium  
4841 'na-9x12-envelope': Specifies the North American 9x12 envelope medium  
4842 'monarch-envelope': Specifies the Monarch envelope  
4843 'na-number-10-envelope': Specifies the North American number 10 business envelope medium

4844 'na-7x9-envelope': Specifies the North American 7x9 inch envelope  
4845 'na-9x11-envelope': Specifies the North American 9x11 inch envelope  
4846 'na-10x14-envelope': Specifies the North American 10x14 inch envelope  
4847 'na-number-9-envelope': Specifies the North American number 9 business envelope  
4848 'na-6x9-envelope': Specifies the North American 6x9 inch envelope  
4849 'na-10x15-envelope': Specifies the North American 10x15 inch envelope  
4850

4851 The following standard values are defined for the less commonly used media (white-only):

4852 'executive-white': Specifies the white executive medium  
4853 'folio-white': Specifies the folio white medium  
4854 'invoice-white': Specifies the white invoice medium  
4855 'ledger-white': Specifies the white ledger medium  
4856 'quarto-white': Specifies the white quarto medium  
4857 'iso-a0-white': Specifies the ISO A0 white medium  
4858 'iso-a1-white': Specifies the ISO A1 white medium  
4859 'iso-a2-white': Specifies the ISO A2 white medium  
4860 'iso-a6-white': Specifies the ISO A6 white medium  
4861 'iso-a7-white': Specifies the ISO A7 white medium  
4862 'iso-a8-white': Specifies the ISO A8 white medium  
4863 'iso-a9-white': Specifies the ISO A9 white medium  
4864 'iso-10-white': Specifies the ISO A10 white medium  
4865 'iso-b0-white': Specifies the ISO B0 white medium  
4866 'iso-b1-white': Specifies the ISO B1 white medium  
4867 'iso-b2-white': Specifies the ISO B2 white medium  
4868 'iso-b3-white': Specifies the ISO B3 white medium  
4869 'iso-b6-white': Specifies the ISO B6 white medium  
4870 'iso-b7-white': Specifies the ISO B7 white medium  
4871 'iso-b8-white': Specifies the ISO B8 white medium  
4872 'iso-b9-white': Specifies the ISO B9 white medium  
4873 'iso-b10-white': Specifies the ISO B10 white medium  
4874 'jis-b0-white': Specifies the JIS B0 white medium  
4875 'jis-b1-white': Specifies the JIS B1 white medium  
4876 'jis-b2-white': Specifies the JIS B2 white medium  
4877 'jis-b3-white': Specifies the JIS B3 white medium  
4878 'jis-b6-white': Specifies the JIS B6 white medium  
4879 'jis-b7-white': Specifies the JIS B7 white medium  
4880 'jis-b8-white': Specifies the JIS B8 white medium  
4881 'jis-b9-white': Specifies the JIS B9 white medium  
4882 'jis-b10-white': Specifies the JIS B10 white medium

4883

4884 The following standard values are defined for engineering media:

4885 'a': Specifies the engineering A size medium

4886 'b': Specifies the engineering B size medium

4887 'c': Specifies the engineering C size medium

4888 'd': Specifies the engineering D size medium

4889 'e': Specifies the engineering E size medium

4890

4891 The following standard values are defined for input-trays (from ISO DPA and the Printer MIB):

4892 'top': The top input tray in the printer.

4893 'middle': The middle input tray in the printer.

4894 'bottom': The bottom input tray in the printer.

4895 'envelope': The envelope input tray in the printer.

4896 'manual': The manual feed input tray in the printer.

4897 'large-capacity': The large capacity input tray in the printer.

4898 'main': The main input tray

4899 'side': The side input tray

4900

4901 The following standard values are defined for media sizes (from ISO DPA):

4902 'iso-a0': Specifies the ISO A0 size: 841 mm by 1189 mm as defined in ISO 216

4903 'iso-a1': Specifies the ISO A1 size: 594 mm by 841 mm as defined in ISO 216

4904 'iso-a2': Specifies the ISO A2 size: 420 mm by 594 mm as defined in ISO 216

4905 'iso-a3': Specifies the ISO A3 size: 297 mm by 420 mm as defined in ISO 216

4906 'iso-a4': Specifies the ISO A4 size: 210 mm by 297 mm as defined in ISO 216

4907 'iso-a5': Specifies the ISO A5 size: 148 mm by 210 mm as defined in ISO 216

4908 'iso-a6': Specifies the ISO A6 size: 105 mm by 148 mm as defined in ISO 216

4909 'iso-a7': Specifies the ISO A7 size: 74 mm by 105 mm as defined in ISO 216

4910 'iso-a8': Specifies the ISO A8 size: 52 mm by 74 mm as defined in ISO 216

4911 'iso-a9': Specifies the ISO A9 size: 37 mm by 52 mm as defined in ISO 216

4912 'iso-a10': Specifies the ISO A10 size: 26 mm by 37 mm as defined in ISO 216

4913 'iso-b0': Specifies the ISO B0 size: 1000 mm by 1414 mm as defined in ISO 216

4914 'iso-b1': Specifies the ISO B1 size: 707 mm by 1000 mm as defined in ISO 216

4915 'iso-b2': Specifies the ISO B2 size: 500 mm by 707 mm as defined in ISO 216

4916 'iso-b3': Specifies the ISO B3 size: 353 mm by 500 mm as defined in ISO 216

4917 'iso-b4': Specifies the ISO B4 size: 250 mm by 353 mm as defined in ISO 216

4918 'iso-b5': Specifies the ISO B5 size: 176 mm by 250 mm as defined in ISO 216

4919 'iso-b6': Specifies the ISO B6 size: 125 mm by 176 mm as defined in ISO 216  
4920 'iso-b7': Specifies the ISO B7 size: 88 mm by 125 mm as defined in ISO 216  
4921 'iso-b8': Specifies the ISO B8 size: 62 mm by 88 mm as defined in ISO 216  
4922 'iso-b9': Specifies the ISO B9 size: 44 mm by 62 mm as defined in ISO 216  
4923 'iso-b10': Specifies the ISO B10 size: 31 mm by 44 mm as defined in ISO 216  
4924 'na-letter': Specifies the North American letter size: 8.5 inches by 11 inches  
4925 'na-legal': Specifies the North American legal size: 8.5 inches by 14 inches  
4926 'executive': Specifies the executive size (7.25 X 10.5 in)  
4927 'folio': Specifies the folio size (8.5 X 13 in)  
4928 'invoice': Specifies the invoice size (5.5 X 8.5 in)  
4929 'ledger': Specifies the ledger size (11 X 17 in)  
4930 'quarto': Specifies the quarto size (8.5 X 10.83 in)  
4931 'iso-c3': Specifies the ISO C3 size: 324 mm by 458 mm as defined in ISO 269  
4932 'iso-c4': Specifies the ISO C4 size: 229 mm by 324 mm as defined in ISO 269  
4933 'iso-c5': Specifies the ISO C5 size: 162 mm by 229 mm as defined in ISO 269  
4934 'iso-c6': Specifies the ISO C6 size: 114 mm by 162 mm as defined in ISO 269  
4935 'iso-designated-long': Specifies the ISO Designated Long size: 110 mm by 220 mm as defined in ISO  
4936 269  
4937 'na-10x13-envelope': Specifies the North American 10x13 size: 10 inches by 13 inches  
4938 'na-9x12-envelope': Specifies the North American 9x12 size: 9 inches by 12 inches  
4939 'na-number-10-envelope': Specifies the North American number 10 business envelope size: 4.125  
4940 inches by 9.5 inches  
4941 'na-7x9-envelope': Specifies the North American 7x9 inch envelope size  
4942 'na-9x11-envelope': Specifies the North American 9x11 inch envelope size  
4943 'na-10x14-envelope': Specifies the North American 10x14 inch envelope size  
4944 'na-number-9-envelope': Specifies the North American number 9 business envelope size  
4945 'na-6x9-envelope': Specifies the North American 6x9 envelope size  
4946 'na-10x15-envelope': Specifies the North American 10x15 envelope size  
4947 'monarch-envelope': Specifies the Monarch envelope size (3.87 x 7.5 in)  
4948 'jis-b0': Specifies the JIS B0 size: 1030mm x 1456mm  
4949 'jis-b1': Specifies the JIS B1 size: 728mm x 1030mm  
4950 'jis-b2': Specifies the JIS B2 size: 515mm x 728mm  
4951 'jis-b3': Specifies the JIS B3 size: 364mm x 515mm  
4952 'jis-b4': Specifies the JIS B4 size: 257mm x 364mm  
4953 'jis-b5': Specifies the JIS B5 size: 182mm x 257mm  
4954 'jis-b6': Specifies the JIS B6 size: 128mm x 182mm  
4955 'jis-b7': Specifies the JIS B7 size: 91mm x 128mm  
4956 'jis-b8': Specifies the JIS B8 size: 64mm x 91mm  
4957 'jis-b9': Specifies the JIS B9 size: 45mm x 64mm  
4958 'jis-b10': Specifies the JIS B10 size: 32mm x 45mm

## 4959 16. APPENDIX D: Processing IPP Attributes

4960 When submitting a print job to a Printer object, the IPP model allows a client to supply operation and Job  
4961 Template attributes along with the document data. These Job Template attributes in the create request  
4962 affect the rendering, production and finishing of the documents in the job. Similar types of instructions  
4963 may also be contained in the document to be printed, that is, embedded within the print data itself. In  
4964 addition, the Printer has a set of attributes that describe what rendering and finishing options which are  
4965 supported by that Printer. This model, which allows for flexibility and power, also introduces the  
4966 potential that at job submission time, these client-supplied attributes may conflict with either:

- 4967 - what the implementation is capable of realizing (i.e., what the Printer supports), as well as
- 4968 - the instructions embedded within the print data itself.

4969  
4970 The following sections describe how these two types of conflicts are handled in the IPP model.

## 4971 16.1 Fidelity

4972 If there is a conflict between what the client requests and what a Printer object supports, the client may  
4973 request one of two possible conflict handling mechanisms:

- 4974 1) either reject the job since the job can not be processed exactly as specified, or
- 4975 2) allow the Printer to make any changes necessary to proceed with processing the Job the best it can.

4976  
4977 In the first case the client is indicating to the Printer object: "Print the job exactly as specified with no  
4978 exceptions, and if that can't be done, don't even bother printing the job at all." In the second case, the  
4979 client is indicating to the Printer object: "It is more important to make sure the job is printed rather than  
4980 be processed exactly as specified; just make sure the job is printed even if client supplied attributes need  
4981 to be changed or ignored."

4982 The IPP model accounts for this situation by introducing an "ipp-attribute-fidelity" attribute.

4983 In a create request, "ipp-attribute-fidelity" is a boolean operation attribute that is OPTIONALLY  
4984 supplied by the client. The value 'true' indicates that total fidelity to client supplied Job Template  
4985 attributes and values is required. The client is requesting that the Job be printed exactly as specified, and  
4986 if that is not possible then the job MUST be rejected rather than processed incorrectly. The value 'false'  
4987 indicates that a reasonable attempt to print the Job is acceptable. If a Printer does not support some of  
4988 the client supplied Job Template attributes or values, the Printer **SHALLMUST** ignore them or substitute  
4989 any supported value for unsupported values, respectively. The Printer may choose to substitute the  
4990 default value associated with that attribute, or use some other supported value that is similar to the  
4991 unsupported requested value. For example, if a client supplies a "media" value of 'na-letter', the Printer

4992 may choose to substitute 'iso-a4' rather than a default value of 'envelope'. If the client does not supply the  
4993 "ipp-attribute-fidelity" attribute, the Printer assumes a value of 'false'.

4994 Each Printer implementation **MUST** support both types of "fidelity" printing (that is whether the client  
4995 supplies a value of 'true' or 'false'):

- 4996 - If the client supplies 'false' or does not supply the attribute, the Printer object **SHALLMUST** always  
4997 accept the request by ignoring unsupported Job Template attributes and by substituting  
4998 unsupported values of supported Job Template attributes with supported values.
- 4999 - If the client supplies 'true', the Printer object **SHALLMUST** reject the request if the client supplies  
5000 unsupported Job Template attributes.

5001

5002 Since a client can always query a Printer to find out exactly what is and is not supported, "ipp-attribute-  
5003 fidelity" set to 'false' is useful when:

- 5004 1) The End-User uses a command line interface to request attributes that might not be supported.
- 5005 2) In a GUI context, if the End User expects the job might be moved to another printer and prefers a  
5006 sub-optimal result to nothing at all.
- 5007 3) The End User just wants something reasonable in lieu of nothing at all.

5008

## 5009 16.2 Page Description Language (PDL) Override

5010 If there is a conflict between the value of an IPP Job Template attribute and a corresponding instruction  
5011 in the document data, the value of the IPP attribute **SHOULD** take precedence over the document  
5012 instruction. Consider the case where a previously formatted file of document data is sent to an IPP  
5013 Printer. In this case, if the client supplies any attributes at job submission time, the client desires that  
5014 those attributes override the embedded instructions. Consider the case were a previously formatted  
5015 document has embedded in it commands to load 'iso-a4' media. However, the document is passed to an  
5016 end user that only has access to a printer with 'na-letter' media loaded. That end user most likely wants to  
5017 submit that document to an IPP Printer with the "media" Job Template attribute set to 'na-letter'. The job  
5018 submission attribute should take precedence over the embedded PDL instruction. However, until  
5019 companies that supply document data interpreters allow a way for external IPP attributes to take  
5020 precedence over embedded job production instructions, a Printer might not be able to support the  
5021 semantics that IPP attributes override the embedded instructions.

5022 The IPP model accounts for this situation by introducing a "pdl-override-supported" attribute that  
5023 describes the Printer objects capabilities to override instructions embedded in the PDL data stream. The  
5024 value of the "pdl-override-supported" attribute is configured by means outside IPP/1.0.

5025 This **MANDATORYREQUIRED** Printer attribute takes on the following values:

- 5026 - 'attempted': This value indicates that the Printer object attempts to make the IPP attribute values  
5027 take precedence over embedded instructions in the document data, however there is no guarantee.  
5028 - 'not-attempted': This value indicates that the Printer object makes no attempt to make the IPP  
5029 attribute values take precedence over embedded instructions in the document data.  
5030

5031 At job processing time, an implementation that supports the value of 'attempted' might do one of several  
5032 different actions:

- 5033 1) Generate an output device specific command sequence to realize the feature represented by the IPP  
5034 attribute value.
- 5035 2) Parse the document data itself and replace the conflicting embedded instruction with a new  
5036 embedded instruction that matches the intent of the IPP attribute value.
- 5037 3) Indicate to the Printer that external supplied attributes take precedence over embedded instructions  
5038 and then pass the external IPP attribute values to the document data interpreter.
- 5039 4) Anything else that allows for the semantics that IPP attributes override embedded document data  
5040 instructions.

5041  
5042 Since 'attempted' does not offer any type of guarantee, even though a given Printer object might not do a  
5043 very "good" job of attempting to ensure that IPP attributes take a higher precedence over instructions  
5044 embedded in the document data, it would still be a conforming implementation.

5045 At job processing time, an implementation that supports the value of 'not-attempted' might do one of the  
5046 following actions:

- 5047 1) Simply pre-pend the document data with the PDL instruction that corresponds to the client-  
5048 supplied PDL attribute, such that if the document data also has the same PDL instruction, it will  
5049 override what the Printer object pre-pended. In other words, this implementation is using the  
5050 same implementation semantics for the client-supplied IPP attributes as for the Printer object  
5051 defaults.
- 5052 2) Parse the document data and replace the conflicting embedded instruction with a new embedded  
5053 instruction that approximates, but does not match, the semantic intent of the IPP attribute value.  
5054

5055 Note: The "ipp-attribute-fidelity" attribute applies to the Printer's ability to either accept or reject other  
5056 unsupported Job Template attributes. In other words, if "ipp-attribute-fidelity" is set to 'true', a Job is  
5057 accepted if and only if the client supplied Job Template attributes and values are supported by the Printer.  
5058 Whether these attributes actually affect the processing of the Job when the document data contains  
5059 embedded instructions depends on the ability of the Printer to override the instructions embedded in the  
5060 document data with the semantics of the IPP attributes. If the document data attributes can be  
5061 overridden ("pdl-override-supported" set to 'attempted'), the Printer makes an attempt to use the IPP  
5062 attributes when processing the Job. If the document data attributes can not be overridden ("pdl-override-  
5063 supported" set to 'not-attempted'), the Printer makes no attempt to override the embedded document data



5064 instructions with the IPP attributes when processing the Job, and hence, the IPP attributes may fail to  
5065 affect the Job processing and output when the corresponding instruction is embedded in the document  
5066 data.

### 5067 16.3 Suggested Operation Processing Steps for All Operations

5068 When an IPP object receives a request, the IPP object either accepts or rejects the request. In order to  
5069 determine whether or not to accept or reject the request, the IPP object SHOULD execute the following  
5070 steps. The order of the steps may be rearranged and/or combined, including making one or multiple  
5071 passes over the request. Therefore, the error status codes returned may differ between implementations.  
5072 The next section contains the additional steps for the Print-Job, Validate-Job, Print-URI, Create-Job,  
5073 Send-Document, and Send-URI operations that create jobs, adds documents, and validates jobs.

5074 In the following, processing continues step by step until a "RETURNS the xxx status code ..." statement  
5075 is encountered. Error returns are indicated by the verb: "REJECTS". Since clients have difficulty getting  
5076 the status code before sending all of the document data in a Print-Job request, clients SHOULD use the  
5077 Validate-Job operation before sending large documents to be printed, in order to validate whether the IPP  
5078 Printer will accept the job or not.

5079 It is assumed that security authentication and authorization has already taken place at a lower layer.

#### 5080 16.3.1 Validate version number

5081 Every request and every response contains the "version-number" attribute. The value of this attribute is  
5082 the major and minor version number of the syntax and semantics that the client and IPP object is using,  
5083 respectively. The "version-number" attribute remains in a fixed position across all future versions so that  
5084 all clients and IPP object that support future versions can determine which version is being used. The IPP  
5085 object checks to see if the major version number supplied in the request is supported. If not, the Printer  
5086 object REJECTS the request and RETURNS the 'server-error-version-not-supported' status code in the  
5087 response. The IPP object returns in the "version-number" response attribute the major and minor version  
5088 for the error response. Thus the client can learn at least one major and minor version that the IPP object  
5089 supports. The IPP object is encouraged to return the closest version number to the one supplied by the  
5090 client.

5091 The checking of the minor version number is implementation dependent, however if the client supplied  
5092 minor version is explicitly supported, the IPP object ~~SHALL~~**MUST** respond using that identical minor  
5093 version number. If the requested minor version is not supported (the requested minor version is either  
5094 higher or lower) than a supported minor version, the IPP object SHOULD return the closest supported  
5095 minor version.

## 5096 16.3.2 Validate operation identifier

5097 The Printer object checks to see if the "operation-id" attribute supplied by the client is supported as  
5098 indicated in the Printer object's "printer-operations-supported" attribute. If not, the Printer REJECTS the  
5099 request and returns the 'server-error-operation-not-supported' status code in the response.

## 5100 16.3.3 Validate the request identifier

5101 The Printer object checks to see if the "request-id" attribute supplied by the client is in range. If the value  
5102 is not between 1 and  $2^{*}31 - 1$  (inclusive), the Printer object REJECTS the request and returns the  
5103 'client-error-bad-request' status code in the response.

5104 Note: The "version-number", attribute, "operation-id", and the "request-id" attributes in the same fixed  
5105 octet positions in all versions of the protocol. These fields are validated before proceeding with the rest  
5106 of the validation.

## 5107 16.3.4 Validate attribute group and attribute presence and order

5108 The order of the following validation steps depends on implementation.

## 5109 16.3.4.1 Validate the presence and order of attribute groups

5110 Client requests and IPP object responses contain attribute groups that Section 3 requires to be present  
5111 and in a specified order. An IPP object verifies that the attribute groups are present and in the correct  
5112 order in requests supplied by clients (attribute groups without an \* in the following tables).

5113 If an IPP object receives a request with (1) required attribute groups missing, or (2) the attributes groups  
5114 are out of order, or (3) the groups are repeated, the IPP object REJECTS the request and RETURNS the  
5115 'client-error-bad-request' status code. For example, it is an error for the Job Template Attributes group  
5116 to occur before the Operation Attributes group, for the Operation Attributes group to be omitted, or for  
5117 an attribute group to occur more than once, except in the Get-Jobs response.

5118 Since this kind of attribute group error is most likely to be an error detected by a client developer rather  
5119 than by a customer, the IPP object NEED NOT return an indication of which attribute group was in error  
5120 in either the Unsupported Attributes group or the Status Message. Also, the IPP object NEED NOT find  
5121 all attribute group errors before returning this error.

## 5122 16.3.4.2 Ignore unknown attribute groups in the expected position

5123 Future attribute groups may be added to the specification at the end of requests just before the Document  
5124 Content and at the end of response, except for the Get-Jobs response, where it maybe there or before the

5125 first job attributes returned. If an IPP object receives an unknown attribute group in these positions, it  
5126 ignores the entire group, rather than returning an error, since that group may be a new group in a later  
5127 minor version of the protocol that can be ignored. (If the new attribute group cannot be ignored without  
5128 confusing the client, the major version number would have been increased in the protocol document and  
5129 in the request). If the unknown group occurs in a different position, the IPP object REJECTS the request  
5130 and RETURNS the 'client-error-bad-request' status code.

5131 Clients also ignore unknown attribute groups returned in a response.

5132 Note: By validating that requests are in the proper form, IPP objects force clients to use the proper form  
5133 which, in turn, increases the chances that customers will be able to use such clients from multiple vendors  
5134 with IPP objects from other vendors.

#### 5135 16.3.4.3 Validate the presence of a single occurrence of required Operation attributes

5136 Client requests and IPP object responses contain Operation attributes that Section 3 requires to be  
5137 present. Attributes within a group may be in any order, except for the ordering of target, charset, and  
5138 natural languages attributes. These attributes must be first, and must be supplied in the following order:  
5139 charset, natural language, and then target. An IPP object verifies that the attributes that Section 4  
5140 requires to be supplied by the client have been supplied in the request (attributes without an \* in the  
5141 following tables). An asterisk (\*) indicates groups and Operation attributes that the client may omit in a  
5142 request or an IPP object may omit in a response.

5143 If an IPP object receives a request with required attributes missing or repeated from a group, the IPP  
5144 object REJECTS the request and RETURNS the 'client-error-bad-request' status code. For example, it is  
5145 an error for the "attributes-charset" or "attributes-natural-language" attribute to be omitted in any  
5146 operation request, or for an Operation attribute to be supplied in a Job Template group or a Job Template  
5147 attribute to be supplied in an Operation Attribute group in a create request. It is also an error to supply  
5148 the "attributes-charset" attribute twice.

5149 Since these kinds of attribute errors are most likely to be detected by a client developer rather than by a  
5150 customer, the IPP object NEED NOT return an indication of which attribute was in error in either the  
5151 Unsupported Attributes group or the Status Message. Also, the IPP object NEED NOT find all attribute  
5152 errors before returning this error.

5153 The following tables list all the attributes for all the operations by attribute group in each request and  
5154 each response. The order of the groups is the order that the client supplies the groups as specified in  
5155 Section 3. The order of the attributes within a group is arbitrary, except as noted for some of the special  
5156 operation attributes (charset, natural language, and target). The tables below use the following notation:

5157 **MR** indicates a **MANDATORYREQUIRED** attribute that an IPP object MUST support

5158 O indicates an OPTIONAL attribute that an IPP object NEED NOT support  
 5159 \* indicates that a client MAY omit the attribute in a request and that an IPP object MAY  
 5160 omit the attribute in a response. The absence of an \* means that a client MUST  
 5161 supply the attribute in a request and an IPP object MUST supply the attribute in a  
 5162 response.

### 5164 Operation Requests

5165 The tables below show the attributes in their proper attribute groups for operation requests:

5166 Note: All operation requests contain ~~the following common elements:~~  
 5167 "version-number", "operation-id", and "request-id" parameters.

5168 Print-Job Request:

5170 Group 1: Operation Attributes (MR)  
 5171 attributes-charset (MR)  
 5172 attributes-natural-language (MR)  
 5173 printer-uri (MR)  
 5174 requesting-user-name (MR\*)  
 5175 job-name (MR\*)  
 5176 ipp-attribute-fidelity (MR\*)  
 5177 document-name (MR\*)  
 5178 document-format (MR\*)  
 5179 document-natural-language (O\*)  
 5180 compression (O\*)  
 5181 job-k-octets (O\*)  
 5182 job-impressions (O\*)  
 5183 job-media-sheets (O\*)  
 5184 Group 2: Job Template Attributes (MR)  
 5185 <Job Template attributes> (O\*) (see Section 4.2)  
 5186 Group 3: Document Content (MR)  
 5187 <document content>

5188 Validate-Job Request:

5190 Group 1: Operation Attributes (MR)  
 5191 attributes-charset (MR)  
 5192 attributes-natural-language (MR)  
 5193 printer-uri (MR)  
 5194 requesting-user-name (MR\*)  
 5195 job-name (MR\*)  
 5196 ipp-attribute-fidelity (MR\*)  
 5197 document-name (MR\*)  
 5198 document-format (MR\*)  
 5199 document-natural-language (O\*)

5200           compression (O\*)  
5201           job-k-octets (O\*)  
5202           job-impressions (O\*)  
5203           job-media-sheets (O\*)  
5204       Group 2: Job Template Attributes (MR)  
5205           <Job Template attributes> (O\*) (see Section 4.2)  
5206  
5207   Create-Job Request:  
5208       Group 1: Operation Attributes (MR)  
5209           attributes-charset (MR)  
5210           attributes-natural-language (MR)  
5211           printer-uri (MR)  
5212           requesting-user-name (MR\*)  
5213           job-name (MR\*)  
5214           ipp-attribute-fidelity (MR\*)  
5215           job-k-octets (O\*)  
5216           job-impressions (O\*)  
5217           job-media-sheets (O\*)  
5218       Group 2: Job Template Attributes (MR)  
5219           <Job Template attributes> (O\*) (see Section 4.2)  
5220  
5221   Print-URI Request:  
5222       Group 1: Operation Attributes (MR)  
5223           attributes-charset (MR)  
5224           attributes-natural-language (MR)  
5225           printer-uri (MR)  
5226           document-uri (MR)  
5227           requesting-user-name (MR\*)  
5228           job-name (MR\*)  
5229           ipp-attribute-fidelity (MR\*)  
5230           document-name (MR\*)  
5231           document-format (MR\*)  
5232           document-natural-language (O\*)  
5233           compression (O\*)  
5234           job-k-octets (O\*)  
5235           job-impressions (O\*)  
5236           job-media-sheets (O\*)  
5237       Group 2: Job Template Attributes (MR)  
5238           <Job Template attributes> (O\*) (see Section 4.2)  
5239  
5240   Send-Document Request:  
5241       Group 1: Operation Attributes (MR)  
5242           attributes-charset (MR)  
5243           attributes-natural-language (MR)  
5244           (printer-uri & job-id) | job-uri (MR)

5245           last-document (MR)  
5246           requesting-user-name (MR\*)  
5247           document-name (MR\*)  
5248           document-format (MR\*)  
5249           document-natural-language (O\*)  
5250           compression (O\*)  
5251       Group 2: Document Content (MR)  
5252           <document content>  
5253  
5254   Send-URI Request:  
5255       Group 1: Operation Attributes (MR)  
5256           attributes-charset (MR)  
5257           attributes-natural-language (MR)  
5258           (printer-uri & job-id) | job-uri (MR)  
5259           last-document (MR)  
5260           document-uri (MR)  
5261           requesting-user-name (MR\*)  
5262           document-name (MR\*)  
5263           document-format (MR\*)  
5264           document-natural-language (O\*)  
5265           compression (O\*)  
5266  
5267   Cancel-Job Request:  
5268       Group 1: Operation Attributes (MR)  
5269           attributes-charset (MR)  
5270           attributes-natural-language (MR)  
5271           (printer-uri & job-id) | job-uri (MR)  
5272           requesting-user-name (MR\*)  
5273           message (O\*)  
5274  
5275   Get-Printer-Attributes Request:  
5276       Group 1: Operation Attributes (MR)  
5277           attributes-charset (MR)  
5278           attributes-natural-language (MR)  
5279           printer-uri (MR)  
5280           requesting-user-name (MR\*)  
5281           requested-attributes (MR\*)  
5282           document-format (MR\*)  
5283  
5284   Get-Job-Attributes Request:  
5285       Group 1: Operation Attributes (MR)  
5286           attributes-charset (MR)  
5287           attributes-natural-language (MR)  
5288           (printer-uri & job-id) | job-uri (MR)  
5289           requesting-user-name (MR\*)

5290 requested-attributes (MR\*)  
 5291  
 5292 Get-Jobs Request:  
 5293 Group 1: Operation Attributes (MR)  
 5294 attributes-charset (MR)  
 5295 attributes-natural-language (MR)  
 5296 printer-uri (MR)  
 5297 requesting-user-name (MR\*)  
 5298 limit (MR\*)  
 5299 requested-attributes (MR\*)  
 5300 which-jobs (MR\*)  
 5301 my-jobs (MR\*)  
 5302

5303 Operation Responses

5304 The tables below show the response attributes in their proper attribute groups for responses.

5305 Note: All operation responses contain ~~the following common elements:~~  
 5306 "version-number", "status-code", and "request-id" parameters.  
 5307

5308 Print-Job Response:

5309 Print-URI Response:

5310 Create-Job Response:

5311 Send-Document Response:

5312 Send-URI Response:

5313 Group 1: Operation Attributes (MR)  
 5314 attributes-charset (MR)  
 5315 attributes-natural-language (MR)  
 5316 status-message (0\*)  
 5317 Group 2: Unsupported Attributes (MR\*) (see Note 3)  
 5318 <unsupported attributes> (MR\*)  
 5319 Group 3: Job Object Attributes(MR\*) (see Note 2)  
 5320 job-uri (MR)  
 5321 job-id (MR)  
 5322 job-state (MR)  
 5323 job-state-reasons (0\*)  
 5324 job-state-message (0\*)  
 5325 number-of-intervening-jobs (0\*)  
 5326

5327 Validate-Job Response:

5328 Cancel-Job Response:

5329 Group 1: Operation Attributes (MR)  
 5330 attributes-charset (MR)

5331           attributes-natural-language (MR)  
5332           status-message (O\*)  
5333       Group 2: Unsupported Attributes (MR\*) (see Note 3)  
5334           <unsupported attributes> (MR\*)

5335

5336 Note 2 - the Job Object Attributes and Printer Object Attributes are  
5337 returned only if the IPP object returns one of the success status  
5338 codes.

5339

5340 Note 3 - the Unsupported Attributes Group is present only if the  
5341 client included some Operation and/or Job Template attributes that the  
5342 Printer doesn't support whether a success or an error return.

5343

5344 Get-Printer-Attributes Response:

5345       Group 1: Operation Attributes (MR)  
5346           attributes-charset (MR)  
5347           attributes-natural-language (MR)  
5348           status-message (O\*)  
5349       Group 2: Unsupported Attributes (MR\*) (see Note 4)  
5350           <unsupported attributes> (MR\*)  
5351       Group 3: Printer Object Attributes(MR\*) (see Note 2)  
5352           <requested attributes> (MR\*)

5353

5354 Note 4 - the Unsupported Attributes Group is present only if the  
5355 client included some Operation attributes that the Printer doesn't  
5356 support whether a success or an error return.

5357

5358 Get-Job-Attributes Response:

5359       Group 1: Operation Attributes (MR)  
5360           attributes-charset (MR)  
5361           attributes-natural-language (MR)  
5362           status-message (O\*)  
5363       Group 2: Unsupported Attributes (MR\*) (see Note 4)  
5364           <unsupported attributes> (MR\*)  
5365       Group 3: Job Object Attributes(MR\*) (see Note 2)  
5366           <requested attributes> (MR\*)

5367

5368 Get-Jobs Response:

5369       Group 1: Operation Attributes (MR)  
5370           attributes-charset (MR)  
5371           attributes-natural-language (MR)  
5372           status-message (O\*)  
5373       Group 2: Unsupported Attributes (MR\*) (see Note 4)  
5374           <unsupported attributes> (MR\*)  
5375       Group 3: Job Object Attributes(MR\*) (see Note 2, 5)



5376 <requested attributes> (MR\*)

5377

5378 Note 5: for the Get-Jobs operation the response contains a separate  
5379 Job Object Attributes group 3 to N containing requested-attributes for  
5380 each job object in the response.

5381

5382 16.3.5 Validate the values of the MANDATORYREQUIRED Operation attributes

5383 An IPP object validates the values supplied by the client of the MANDATORYREQUIRED Operation  
5384 attribute that the IPP object MUST support. The next section specifies the validation of the values of the  
5385 OPTIONAL Operation attributes that IPP objects MAY support.

5386 The IPP object performs the following syntactic validation checks of each Operation attribute value:

5387

- 5388 a) that the length of each Operation attribute value is correct for the attribute syntax tag supplied
- 5389 by the client according to Section 4.1.
- 5390 b) that the attribute syntax tag is correct for that Operation attribute according to Section 3,
- 5391 c) that the value is in the range specified for that Operation attribute according to Section 3,
- 5392 d) that multiple values are supplied by the client only for operation attributes that are multi-
- 5393 valued, i.e., that are 1setOf X according to Section 3.

5394

5395 If any of these checks fail, the IPP object REJECTS the request and RETURNS the 'client-error-bad-  
5396 request' or the 'client-error-request-value-too-long' status code. Since such an error is most likely to be  
5397 an error detected by a client developer, rather than by an end-user, the IPP object NEED NOT return an  
5398 indication of which attribute had the error in either the Unsupported Attributes Group or the Status  
5399 Message. The description for each of these syntactic checks is explicitly expressed in the first IF  
5400 statement in the following table.

5401 In addition, the IPP object checks each Operation attribute value against some Printer object attribute or  
5402 some hard-coded value if there is no "xxx-supported" Printer object attribute defined. If its value is not  
5403 among those supported or is not in the range supported, then the IPP object REJECTS the request and  
5404 RETURNS the error status code indicated in the table by the second IF statement. If the value of the  
5405 Printer object's "xxx-supported" attribute is 'no-value' (because the system administrator hasn't configured  
5406 a value), the check always fails.

5407

-----

5408 attributes-charset (charset)

5409 IF NOT any single non-empty 'charset' value less than or equal to 63 octets, REJECT/RETURN  
5410 'client-error-request-value-too-long'.

5411 IF NOT in the Printer object's "charset-supported" attribute, REJECT/RETURN "client-error-  
5412 charset-not-supported".

5413  
5414 attributes-natural-language(naturalLanguage)  
5415     IF NOT any single non-empty 'naturalLanguage' value less than or equal to 63 octets,  
5416     REJECT/RETURN 'client-error-request-value-too-long'.  
5417     ACCEPT the request even if not a member of the set in the Printer object's "generated-natural-  
5418     language-supported" attribute.  
5419  
5420 requesting-user-name  
5421     IF NOT any single 'name' value less than or equal to 255 octets, REJECT/RETURN 'client-error-  
5422     request-value-too-long'.  
5423     IF the IPP object can obtain a better authenticated name, use it instead.  
5424  
5425 job-name(name)  
5426     IF NOT any single 'name' value less than or equal to 255 octets, REJECT/RETURN 'client-error-  
5427     request-value-too-long'.  
5428     IF NOT supplied by the client, the Printer object creates a name from the document-name or  
5429     document-uri.  
5430  
5431 document-name (name)  
5432     IF NOT any single 'name' value less than or equal to 255 octets, REJECT/RETURN 'client-error-  
5433     request-value-too-long'.  
5434  
5435 ipp-attribute-fidelity (boolean)  
5436     IF NOT either a single 'true' or 'false' 'boolean' value equal to 1 octet, REJECT/RETURN 'client-  
5437     error-bad-request'.  
5438     IF NOT supplied by the client, the IPP object assumes the value 'false'.  
5439  
5440 document-format (mimeMediaType)  
5441     IF NOT any single non-empty 'mimeMediaType' value less than or equal to 255 octets,  
5442     REJECT/RETURN 'client-error-request-value-too-long'.  
5443     IF NOT in the Printer object's "document-format-supported" attribute, REJECT/RETURN 'client-  
5444     error-document-format-not-supported'.  
5445     IF NOT supplied by the client, the IPP object assumes the value of the Printer object's "document-  
5446     format-default" attribute.  
5447  
5448 document-uri (uri)  
5449     IF NOT any single non-empty 'uri' value less than or equal to 1023 octets, REJECT/RETURN 'client-  
5450     error-request-value-too-long'.  
5451     IF the URI syntax is not valid, REJECT/RETURN 'client-error-bad-request'.

5452 IF scheme is NOT in the Printer object's "reference-uri-schemes-supported" attribute,  
5453 REJECT/RETURN 'client-error'-uri-scheme-not-supported'.  
5454

5455 last-document (boolean)  
5456 IF NOT either a single 'true' or 'false' 'boolean' value equal to 1 octet, REJECT/RETURN 'client-  
5457 error-bad-request'.  
5458

5459 job-id (integer(1:MAX))  
5460 IF NOT any single 'integer' value equal to 4 octets AND in the range 1 to MAX, REJECT/RETURN  
5461 'client-error-bad-request'.  
5462 IF NOT a job-id of an existing Job object, REJECT/RETURN 'client-error-not-found' or 'client-error-  
5463 gone' status code, if keep track of recently deleted jobs.  
5464

5465 requested-attributes (1setOf keyword)  
5466 IF NOT any number of 'keyword' values less than or equal to 255 octets, REJECT/RETURN 'client-  
5467 error-request-value-too-long'.  
5468 Ignore unsupported values which are the keyword names of unsupported attributes. Don't bother to  
5469 copy such requested (unsupported) attributes to the Unsupported Attribute response group since  
5470 the response will not return them.  
5471

5472 which-jobs (type2 keyword)  
5473 IF NOT a single 'keyword' value less than or equal to 255 octets, REJECT/RETURN 'client-error-  
5474 request-value-too-long'.  
5475 IF NEITHER 'completed' NOR 'not-completed', copy the attribute and the unsupported value to the  
5476 Unsupported Attributes response group and REJECT/RETURN 'client-error-attributes-or-values-  
5477 not-supported'.  
5478 Note: a Printer still supports the 'completed' value even if it keeps no completed/canceled/aborted  
5479 jobs: by returning no jobs when so queried.  
5480 IF NOT supplied by the client, the IPP object assumes the 'not-completed' value.  
5481

5482 my-jobs (boolean)  
5483 IF NOT either a single 'true' or 'false' 'boolean' value equal to 1 octet, REJECT/RETURN 'client-  
5484 error-bad-request'.  
5485 IF NOT supplied by the client, the IPP object assumes the 'false' value.  
5486

5487 limit (integer(1:MAX))  
5488 IF NOT any single 'integer' value equal to 4 octets AND in the range 1 to MAX, REJECT/RETURN  
5489 'client-error-bad-request'.  
5490 IF NOT supplied by the client, the IPP object returns all jobs, no matter how many.  
5491

5492 -----  
5493

5494 16.3.6 Validate the values of the OPTIONAL Operation attributes

5495 OPTIONAL Operation attributes are those that an IPP object MAY or MAY NOT support. An IPP  
5496 object validates the values of the OPTIONAL attributes supplied by the client. The IPP object performs  
5497 the same syntactic validation checks for each OPTIONAL attribute value as in Section 16.3.5. As in  
5498 Section 16.3.5, if any fail, the IPP object REJECTS the request and RETURNS the 'client-error-bad-  
5499 request' or the 'client-error-request-value-too-long' status code.

5500 In addition, the IPP object checks each Operation attribute value against some Printer attribute or some  
5501 hard-coded value if there is no "xxx-supported" Printer attribute defined. If its value is not among those  
5502 supported or is not in the range supported, then the IPP object REJECTS the request and RETURNS the  
5503 error status code indicated in the table. If the value of the Printer object's "xxx-supported" attribute is  
5504 'no-value' (because the system administrator hasn't configured a value), the check always fails.

5505 If the IPP object doesn't recognize/support an attribute, the IPP object treats the attribute as an unknown  
5506 or unsupported attribute (see the last row in the table below).

5507 -----  
5508 document-natural-language (naturalLanguage)

5509 IF NOT any single non-empty 'naturalLanguage' value less than or equal to 63 octets,  
5510 REJECT/RETURN 'client-error-request-value-too-long'.

5511 IF NOT a value that the Printer object supports in document formats, (no standard "xxx-supported"  
5512 Printer attribute), REJECT/RETURN 'client-error-natural-language-not-supported'.

5513  
5514 compression (type3 keyword)

5515 IF NOT any single 'keyword' values less than or equal to 255 octets, REJECT/RETURN 'client-error-  
5516 request-value-too-long'.

5517 IF NOT in the Printer object's "compression-supported" attribute, copy the attribute and the  
5518 unsupported value to the Unsupported Attributes response group and REJECT/RETURN 'client-  
5519 error-attributes-or-values-not-supported'.

5520  
5521 job-k-octets (integer(0:MAX))

5522 IF NOT any single 'integer' value equal to 4 octets,  
5523 REJECT/RETURN 'client-error-bad-request'.

5524 IF NOT in the range of the Printer object's "job-k-octets-supported" attribute, copy the attribute and  
5525 the unsupported value to the Unsupported Attributes response group and REJECT/RETURN  
5526 'client-error-attributes-or-values-not-supported'.

5527

5528 job-impressions (integer(0:MAX))  
5529 IF NOT any single 'integer' value equal to 4 octets,  
5530 REJECT/RETURN 'client-error-bad-request'.  
5531 IF NOT in the range of the Printer object's "job-impressions-supported" attribute, copy the attribute  
5532 and the unsupported value to the Unsupported Attributes response group and REJECT/RETURN  
5533 'client-error-attributes-or-values-not-supported'.  
5534

5535 job-media-sheets (integer(0:MAX))  
5536 IF NOT any single 'integer' value equal to 4 octets,  
5537 REJECT/RETURN 'client-error-bad-request'.  
5538 IF NOT in the range of the Printer object's "job-media-supported" attribute, copy the attribute and the  
5539 unsupported value to the Unsupported Attributes response group and REJECT/RETURN 'client-  
5540 error-attributes-or-values-not-supported'.  
5541

5542 message (text(127))  
5543 IF NOT any single 'text' value less than or equal to 127 octets,  
5544 REJECT/RETURN 'client-error-request-value-too-long'.  
5545

5546 unknown or unsupported attribute  
5547 IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute  
5548 syntax, REJECT/RETURN 'client-error-request-value-too-long'.  
5549 ELSE copy the attribute and value to the Unsupported Attributes response group and change the  
5550 attribute value to the "out-of-band" 'unsupported' value, but otherwise ignore the attribute.  
5551

5552 Note: Future Operation attributes may be added to the protocol specification that may occur  
5553 anywhere in the specified group. When the operation is otherwise successful, the IPP object returns  
5554 the 'successful-ok-ignored-or-substituted-attributes' status code. Ignoring unsupported Operation  
5555 attributes in all operations is analogous to the handling of unsupported Job Template attributes in the  
5556 create and Validate-Job operations when the client supplies the "ipp-attribute-fidelity" Operation  
5557 attribute with the 'false' value. This last rule is so that we can add OPTIONAL Operation attributes to  
5558 future versions of IPP so that older clients can inter-work with new IPP objects and newer clients can  
5559 inter-work with older IPP objects. (If the new attribute cannot be ignored without performing  
5560 unexpectedly, the major version number would have been increased in the protocol document and in  
5561 the request). This rule for Operation attributes is independent of the value of the "ipp-attribute-  
5562 fidelity" attribute. For example, if an IPP object doesn't support the OPTIONAL "job-k-octets"  
5563 attribute', the IPP object treats "job-k-octets" as an unknown attribute and only checks the length for  
5564 the 'integer' attribute syntax supplied by the client. If it is not four octets, the IPP object REJECTS  
5565 the request and RETURNS the 'client-error-bad-request' status code, else the IPP object copies the  
5566 attribute to the Unsupported Attribute response group, setting the value to the "out-of-band"  
5567 'unsupported' value, but otherwise ignores the attribute.

5568 -----

5569 16.4 Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add  
5570 Documents

5571 This section in combination with the previous section recommends the processing steps for the Print-Job,  
5572 Validate-Job, Print-URI, Create-Job, Send-Document, and Send-URI operations that IPP objects  
5573 SHOULD use. These are the operations that create jobs, validate a Print-Job request, and add  
5574 documents to a job.

5575 16.4.1 Default "ipp-attribute-fidelity" if not supplied

5576 The Printer object checks to see if the client supplied an "ipp-attribute-fidelity" Operation attribute. If the  
5577 attribute is not supplied by the client, the IPP object assumes that the value is 'false'.

5578 16.4.2 Check that the Printer object is accepting jobs

5579 If the value of the Printer object's "printer-is-accepting-jobs" is 'false', the Printer object REJECTS the  
5580 request and RETURNS the 'server-error-not-accepting-jobs' status code.

5581 16.4.3 Validate the values of the Job Template attributes

5582 An IPP object validates the values of all Job Template attribute supplied by the client. The IPP object  
5583 performs the analogous syntactic validation checks of each Job Template attribute value that it performs  
5584 for Operation attributes (see Section 16.3.5.):

- 5585 a) that the length of each value is correct for the attribute syntax tag supplied by the client  
5586 according to Section 4.1.  
5587 b) that the attribute syntax tag is correct for that attribute according to Sections 4.2 to 4.4,  
5588 c) that multiple values are supplied only for multi-valued attributes, i.e., that are 1setOf X  
5589 according to Sections 4.2 to 4.4  
5590

5591 As in Section 16.3.5, if any of these syntactic checks fail, the IPP object REJECTS the request and  
5592 RETURNS the 'client-error-bad-request' or 'client-error-request-value-too-long' status code, independent  
5593 of the value of the "ipp-attribute-fidelity". Since such an error is most likely to be an error detected by a  
5594 client developer, rather than by an end-user, the IPP object NEED NOT return an indication of which  
5595 attribute had the error in either the Unsupported Attributes Group or the Status Message. The  
5596 description for each of these syntactic checks is explicitly expressed in the first IF statement in the  
5597 following table.

5598 In addition, the IPP object loops through all the client-supplied Job Template attributes, checking to see if  
5599 the supplied attribute value(s) are supported or in the range supported, i.e., the value of the "xxx"  
5600 attribute in the request is (1) a member of the set of values or is in the range of values of the Printer'  
5601 objects "xxx-supported" attribute. If the value of the Printer object's "xxx-supported" attribute is 'no-  
5602 value' (because the system administrator hasn't configured a value), the check always fails. If the check  
5603 fails, the IPP object copies the attribute to the Unsupported Attributes response group with its  
5604 unsupported value. If the attribute contains more than one value, each value is checked and each  
5605 unsupported value is separately copied, while supported values are not copied. If an IPP object doesn't  
5606 recognize/support a Job Template attribute, i.e., there is no corresponding Printer object "xxx-supported"  
5607 attribute, the IPP object treats the attribute as an unknown or unsupported attribute (see the last row in  
5608 the table below).

5609 If some Job Template attributes are supported for some document formats and not for others or the  
5610 values are different for different document formats, the IPP object SHOULD take that into account in  
5611 this validation using the value of the "document-format" supplied by the client (or defaulted to the value  
5612 of the Printer's "document-format-default" attribute, if not supplied by the client). For example, if  
5613 "number-up" is supported for the 'text/plain' document format, but not for the 'application/postscript'  
5614 document format, the check SHOULD (though it NEED NOT) depend on the value of the "document-  
5615 format" operation attribute. See "document-format" in section 3.2.1.1 and 3.2.5.1.

5616 Note: whether the request is accepted or rejected is determined by the value of the "ipp-attribute-fidelity"  
5617 attribute in a subsequent step, so that all Job Template attribute supplied are examined and all  
5618 unsupported attributes and/or values are copied to the Unsupported Attributes response group.

5619 -----

5620 job-priority (integer(1:100))

5621 IF NOT any single 'integer' value equal to 4 octets, REJECT/RETURN 'client-error-bad-request'.

5622 IF NOT supplied by the client, use the value of the Printer object's "job-priority-default" attribute at  
5623 job submission time.

5624 IF NOT in the range 1 to 100, inclusive, copy the attribute and the unsupported value to the  
5625 Unsupported Attributes response group.

5626 Map the value to the nearest supported value in the range 1:100 as specified by the number of  
5627 discrete values indicated by the value of the Printer's "job-priority-supported" attribute. See the  
5628 formula in Section 4.2.1.

5629

5630 job-hold-until (type3 keyword | name)

5631 IF NOT any single 'keyword' or 'name' value less than or equal to 255 octets, REJECT/RETURN  
5632 'client-error-request-value-too-long'.

5633 IF NOT supplied by the client, use the value of the Printer object's "job-hold-until" attribute at job  
5634 submission time.

5635 IF NOT in the Printer object's "job-hold-until-supported" attribute, copy the attribute and the  
5636 unsupported value to the Unsupported Attributes response group.  
5637  
5638 job-sheets (type3 keyword | name)  
5639 IF NOT any single 'keyword' or 'name' value less than or equal to 255 octets, REJECT/RETURN  
5640 'client-error-request-value-too-long'.  
5641 IF NOT in the Printer object's "job-sheets-supported" attribute, copy the attribute and the  
5642 unsupported value to the Unsupported Attributes response group.  
5643  
5644 multiple-document-handling (type2 keyword)  
5645 IF NOT any single 'keyword' value less than or equal to 255 octets, REJECT/RETURN 'client-error-  
5646 request-value-too-long'.  
5647 IF NOT in the Printer object's "multiple-document-handling-supported" attribute, copy the attribute  
5648 and the unsupported value to the Unsupported Attributes response group.  
5649  
5650 copies (integer(1:MAX))  
5651 IF NOT any single 'integer' value equal to 4 octets,  
5652 REJECT/RETURN 'client-error-bad-request'.  
5653 IF NOT in range of the Printer object's "copies-supported" attribute  
5654 copy the attribute and the unsupported value to the Unsupported Attributes response group.  
5655  
5656 finishings (1setOf type2 enum)  
5657 IF NOT any 'enum' value(s) equal to 4 octets, REJECT/RETURN 'client-error-bad-request'.  
5658 IF NOT in the Printer object's "finishings-supported" attribute, copy the attribute and the unsupported  
5659 value(s), but not any supported values, to the Unsupported Attributes response group.  
5660  
5661 page-ranges (1setOf rangeOfInteger(1:MAX))  
5662 IF NOT any 'rangeOfInteger' value(s) each equal to 8 octets, REJECT/RETURN 'client-error-bad-  
5663 request'.  
5664 IF first value is greater than second value in any range, the ranges are not in ascending order, or  
5665 ranges overlap, REJECT/RETURN 'client-error-bad-request'.  
5666 IF the value of the Printer object's "page-ranges-supported" attribute is 'false', copy the attribute to  
5667 the Unsupported Attributes response group and set the value to the "out-of-band" 'unsupported'  
5668 value.  
5669  
5670 sides (type2 keyword)  
5671 IF NOT any single 'keyword' value less than or equal to 255 octets, REJECT/RETURN 'client-error-  
5672 request-value-too-long'.  
5673 IF NOT in the Printer object's "sides-supported" attribute, copy the attribute and the unsupported  
5674 value to the Unsupported Attributes response group.



5675  
5676 number-up (integer(1:MAX))  
5677     IF NOT any single 'integer' value equal to 4 octets,  
5678     REJECT/RETURN 'client-error-bad-request'.  
5679     IF NOT a value or in the range of one of the values of the Printer object's "number-up-supported"  
5680     attribute, copy the attribute and value to the Unsupported Attribute response group.  
5681  
5682 orientation-requested (type2 enum)  
5683     IF NOT any single 'enum' value equal to 4 octets,  
5684     REJECT/RETURN 'client-error-bad-request'.  
5685     IF NOT in the Printer object's "orientation-requested-supported" attribute, copy the attribute and the  
5686     unsupported value to the Unsupported Attributes response group.  
5687  
5688 media (type3 keyword | name)  
5689     IF NOT any single 'keyword' or 'name' value less than or equal to 255 octets, REJECT/RETURN  
5690     'client-error-request-value-too-long'.  
5691     IF NOT in the Printer object's "media-supported" attribute, copy the attribute and the unsupported  
5692     value to the Unsupported Attributes response group.  
5693  
5694 printer-resolution (resolution)  
5695     IF NOT any single 'resolution' value equal to 9 octets,  
5696     REJECT/RETURN 'client-error-bad-request'.  
5697     IF NOT in the Printer object's "multiple-document-handling-supported" attribute, copy the attribute  
5698     and the unsupported value to the Unsupported Attributes response group.  
5699  
5700 print-quality (type2 enum)  
5701     IF NOT any single 'enum' value equal to 4 octets,  
5702     REJECT/RETURN 'client-error-bad-request'.  
5703     IF NOT in the Printer object's "print-quality-supported" attribute, copy the attribute and the  
5704     unsupported value to the Unsupported Attributes response group.  
5705  
5706 unknown or unsupported attribute (i.e., there is no corresponding Printer object "xxx-supported"  
5707 attribute)  
5708     IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute  
5709     syntax,  
5710     REJECT/RETURN 'client-error-bad-request' or 'client-error-request-value-too-long'.  
5711     ELSE copy the attribute and value to the Unsupported Attributes response group and change the  
5712     attribute value to the "out-of-band" 'unsupported' value. Any remaining Job Template Attributes  
5713     are either unknown or unsupported Job Template attributes and are validated algorithmically  
5714     according to their attribute syntax for proper length (see below).

5715 -----

5716  
 5717 If the attribute syntax is supported AND the length check fails, the IPP object REJECTS the request and  
 5718 RETURNS the 'client-error-request-value-too-long' status code, else the IPP object copies the  
 5719 unsupported Job Template attribute to the Unsupported Attributes response group and changes the  
 5720 attribute value to the "out-of-band" 'unsupported' value. The following table shows the length checks for  
 5721 all attribute syntaxes. In the following table: "<=" means less than or equal, "=" means equal to:

5722 Name	Octet length check for read-write attributes
5723 -----	-----
5724 'textWithLanguage	<= 1023 AND 'naturalLanguage' <= 63
5725 'textWithoutLanguage'	<= 1023
5726 'nameWithLanguage'	<= 255 AND 'naturalLanguage' <= 63
5727 'nameWithoutLanguage'	<= 255
5728 'keyword'	<= 255
5729 'enum'	= 4
5730 'uri'	<= 1023
5731 'uriScheme'	<= 63
5732 'charset'	<= 63
5733 'naturalLanguage'	<= 63
5734 'mimeType'	<= 255
5735 'octetString'	<= 1023
5736 'boolean'	= 1
5737 'integer'	= 4
5738 'rangeOfInteger'	= 8
5739 'dateTime'	= 11
5740 'resolution'	= 9
5741 'lsetOf X'	
5742	

#### 5743 16.4.4 Check for conflicting Job Template attributes values

5744 Once all the Operation and Job Template attributes have been checked individually, the Printer object  
 5745 SHOULD check for any conflicting values among all the supported values supplied by the client. For  
 5746 example, a Printer object might be able to staple and to print on transparencies, however due to physical  
 5747 stapling constraints, the Printer object might not be able to staple transparencies. The IPP object copies  
 5748 the supported attributes and their conflicting attribute values to the Unsupported Attributes response  
 5749 group. The Printer object only copies over those attributes that the Printer object either ignores or  
 5750 substitutes in order to resolve the conflict, and it returns the original values which were supplied by the  
 5751 client. For example suppose the client supplies "finishings" equals 'staple' and "media" equals  
 5752 'transparency', but the Printer object does not support stapling transparencies. If the Printer chooses to  
 5753 ignore the stapling request in order to resolve the conflict, the Printer objects returns "finishings" equal to

5754 'staple' in the Unsupported Attributes response group. If any attributes are multi-valued, only the  
5755 conflicting values of the attributes are copied.

5756 Note: The decisions made to resolve the conflict (if there is a choice) is implementation dependent.

#### 5757 16.4.5 Decide whether to REJECT the request

5758 If there were any unsupported Job Template attributes or unsupported/conflicting Job Template attribute  
5759 values and the client supplied the "ipp-attribute-fidelity" attribute with the 'true' value, the Printer object  
5760 REJECTS the request and return the status code:

5761 (1) 'client-error-conflicting-attributes' status code, if there were any conflicts between attributes  
5762 supplied by the client.

5763 (2) 'client-error-attributes-or-values-not-supported' status code, otherwise.  
5764

5765 Note: Unsupported Operation attributes or values that are returned do not affect the status returned in  
5766 this step. If the unsupported Operation attribute was a serious error, the above already rejected the  
5767 request in a previous step. If control gets to this step with unsupported Operation attributes being  
5768 returned, they are not serious errors.

#### 5769 16.4.6 For the Validate-Job operation, RETURN one of the success status codes

5770 If the requested operation is the Validate-Job operation, the Printer object returns:

5771 (1) the "successful-ok" status code, if there are no unsupported or conflicting Job Template attributes  
5772 or values.

5773 (2) the "successful-ok-conflicting-attributes, if there are any conflicting Job Template attribute or  
5774 values.

5775 (3) the "successful-ok-ignored-or-substituted-attributes, if there are only unsupported Job Template  
5776 attributes or values.  
5777

5778 Note: Unsupported Operation attributes or values that are returned do not affect the status returned in  
5779 this step. If the unsupported Operation attribute was a serious error, the above already rejected the  
5780 request in a previous step. If control gets to this step with unsupported Operation attributes being  
5781 returned, they are not serious errors.

#### 5782 16.4.7 Create the Job object with attributes to support

5783 If "ipp-attribute-fidelity" is set to 'false' (or it was not supplied by the client), the Printer object:

- 5784 (1) creates a Job object, assigns a unique value to the job's "job-uri" and "job-id" attributes, and  
5785 initializes all of the job's other supported Job Description attributes.
- 5786 (2) removes all unsupported attributes from the Job object.
- 5787 (3) for each unsupported value, removes either the unsupported value or substitutes the unsupported  
5788 attribute value with some supported value. If an attribute has no values after removing  
5789 unsupported values from it, the attribute is removed from the Job object (so that the normal  
5790 default behavior at job processing time will take place for that attribute).
- 5791 (4) for each conflicting value, removes either the conflicting value or substitutes the conflicting  
5792 attribute value with some other supported value. If an attribute has no values after removing  
5793 conflicting values from it, the attribute is removed from the Job object (so that the normal default  
5794 behavior at job processing time will take place for that attribute).
- 5795

5796 If there were no attributes or values flagged as unsupported, or the value of "ipp-attribute-fidelity" was  
5797 'false', the Printer object is able to accept the create request and create a new Job object. If the "ipp-  
5798 attribute-fidelity" attribute is set to 'true', the Job Template attributes that populate the new Job object are  
5799 necessarily all the Job Template attributes supplied in the create request. If the "ipp-attribute-fidelity"  
5800 attribute is set to 'false', the Job Template attributes that populate the new Job object are all the client  
5801 supplied Job Template attributes that are supported or that have value substitution. Thus, some of the  
5802 requested Job Template attributes may not appear in the Job object because the Printer object did not  
5803 support those attributes. The attributes that populate the Job object are persistently stored with the Job  
5804 object for that Job. A Get-Job-Attributes operation on that Job object will return only those attributes  
5805 that are persistently stored with the Job object.

5806 Note: All Job Template attributes that are persistently stored with the Job object are intended to be  
5807 "override values"; that is, they that take precedence over whatever other embedded instructions might be  
5808 in the document data itself. However, it is not possible for all Printer objects to realize the semantics of  
5809 "override". End users may query the Printer's "pdl-override-supported" attribute to determine if the  
5810 Printer either attempts or does not attempt to override document data instructions with IPP attributes.

5811 There are some cases, where a Printer supports a Job Template attribute and has an associated default  
5812 value set for that attribute. In the case where a client does not supply the corresponding attribute, the  
5813 Printer does not use its default values to populate Job attributes when creating the new Job object; only  
5814 Job Template attributes actually in the create request are used to populate the Job object. The Printer's  
5815 default values are only used later at Job processing time if no other IPP attribute or instruction embedded  
5816 in the document data is present.

5817 Note: If the default values associated with Job Template attributes that the client did not supply were to  
5818 be used to populate the Job object, then these values would become "override values" rather than  
5819 defaults. If the Printer supports the 'attempted' value of the "pdl-override-supported" attribute, then these  
5820 override values could replace values specified within the document data. This is not the intent of the  
5821 default value mechanism. A default value for an attribute is used only if the create request did not specify

5822 that attribute (or it was ignored when allowed by "ipp-attribute-fidelity" being 'false') and no value was  
5823 provided within the content of the document data.

5824 If the client does not supply a value for some Job Template attribute, and the Printer does not support  
5825 that attribute, as far as IPP is concerned, the result of processing that Job (with respect to the missing  
5826 attribute) is undefined.

#### 5827 16.4.8 Return one of the success status codes

5828 Once the Job object has been created, the Printer object accepts the request and returns to the client:

- 5829 (1) the 'successful-ok' status code, if there are no unsupported or conflicting Job Template attributes  
5830 or values.
- 5831 (2) the 'successful-ok-conflicting-attributes' status code, if there are any conflicting Job Template  
5832 attribute or values.
- 5833 (3) the 'successful-ok-ignored-or-substituted-attributes' status code, if there are only unsupported Job  
5834 Template attributes or values.

5835

5836 Note: Unsupported Operation attributes or values that are returned do not affect the status returned in  
5837 this step. If the unsupported Operation attribute was a serious error, the above already rejected the  
5838 request in a previous step. If control gets to this step with unsupported Operation attributes being  
5839 returned, they are not serious errors.

5840 The Printer object also returns Job status attributes that indicate the initial state of the Job ('pending',  
5841 'pending-held', 'processing', etc.), etc. See Print-Job Response, section 3.2.1.2.

#### 5842 16.4.9 Accept appended Document Content

5843 The Printer object accepts the appended Document Content data and either starts it printing, or spools it  
5844 for later processing.

#### 5845 16.4.10 Scheduling and Starting to Process the Job

5846 The Printer object uses its own configuration and implementation specific algorithms for scheduling the  
5847 Job in the correct processing order. Once the Printer object begins processing the Job, the Printer  
5848 changes the Job's state to 'processing'. If the Printer object supports PDL override (the "pdl-override-  
5849 supported" attribute set to 'attempted'), the implementation does its best to see that IPP attributes take  
5850 precedence over embedded instructions in the document data.

## 5851 16.4.11 Completing the Job

5852 The Printer object continues to process the Job until it can move the Job into the 'completed' state. If an  
5853 Cancel-Job operation is received, the implementation eventually moves the Job into the 'canceled' state.  
5854 If the system encounters errors during processing that do not allow it to progress the Job into a  
5855 completed state, the implementation halts all processing, cleans up any resources, and moves the Job into  
5856 the 'aborted' state.

## 5857 16.4.12 Destroying the Job after completion

5858 Once the Job moves to the 'completed', 'aborted', or 'canceled' state, it is an implementation decision as to  
5859 when to destroy the Job object and release all associated resources. Once the Job has been destroyed, the  
5860 Printer would return either the "client-error-not-found" or "client-error-gone" status codes for operations  
5861 directed at that Job.

5862 Note: the Printer object SHOULD NOT re-use a "job-uri" or "job-id" value for a sufficiently long time  
5863 after a job has been destroyed, so that stale references kept by clients are less likely to access the wrong  
5864 (newer) job.

## 5865 16.4.13 Interaction with "ipp-attribute-fidelity"

5866 Some Printer object implementations may support "ipp-attribute-fidelity" set to 'true' and "pdl-override-  
5867 supported" set to 'attempted' and yet still not be able to realize exactly what the client specifies in the  
5868 create request. This is due to legacy decisions and assumptions that have been made about the role of job  
5869 instructions embedded within the document data and external job instructions that accompany the  
5870 document data and how to handle conflicts between such instructions. The inability to be 100% precise  
5871 about how a given implementation will behave is also compounded by the fact that the two special  
5872 attributes, "ipp-attribute-fidelity" and "pdl-override-supported", apply to the whole job rather than  
5873 specific values for each attribute. For example, some implementations may be able to override almost all  
5874 Job Template attributes except for "number-up".

## 5875 16.5 Using Job Template Attributes During Document Processing.

5876 The Printer object uses some of the Job object's Job Template attributes during the processing of the  
5877 document data associated with that job. These include, but are not limited to, "orientation", "number-  
5878 up", "sides", "media", and "copies". The processing of each document in a Job Object ~~SHALL~~**MUST**  
5879 follow the steps below. These steps are intended only to identify when and how attributes are to be used  
5880 in processing document data and any alternative steps that accomplishes the same effect can be used to  
5881 implement this specification.

- 5882 1. Using the client supplied "document-format" attribute or some form of document format detection  
5883 algorithm (if the value of "document-format" is not specific enough), determine whether or not  
5884 the document data has already been formatted for printing. If the document data has been  
5885 formatted, then go to step 2. Otherwise, the document data ~~SHALL~~**MUST** be formatted. The  
5886 formatting detection algorithm is implementation defined and is not specified by this specification.  
5887 The formatting of the document data uses the "orientation-requested" attribute to determine how  
5888 the formatted print data should be placed on a print-stream page, see section 4.2.10 for the  
5889 details.
- 5890
- 5891 2. The document data is in the form of a print-stream in a known media type. The "page-ranges"  
5892 attribute is used to select, as specified in section 4.2.7, a sub-sequence of the pages in the print-  
5893 stream that are to be processed and images.
- 5894
- 5895 3. The input to this step is a sequence of print-stream pages. This step is controlled by the "number-  
5896 up" attribute. If the value of "number-up" is N, then during the processing of the print-stream  
5897 pages, each N print-stream pages are positioned, as specified in section 4.2.9, to create a single  
5898 impression. If a given document does not have N more print-stream pages, then the completion of  
5899 the impression is controlled by the "multiple-document-handling" attribute as described in section  
5900 4.2.4; when the value of this attribute is 'single-document', the print-stream pages of document  
5901 data from subsequent documents is used to complete the impression.
- 5902
- 5903 The size(scaling), position(translation) and rotation of the print-stream pages on the impression is  
5904 implementation defined. Note that during this process the print-stream pages may be rendered to  
5905 a form suitable for placing on the impression; this rendering is controlled by the values of the  
5906 "printer-resolution" and "print-quality" attributes as described in sections 4.2.12 and 4.2.13. In the  
5907 case N=1, the impression is nearly the same as the print-stream page; the differences would only  
5908 be in the size, position and rotation of the print-stream page and/or any decoration, such as a  
5909 frame to the page, that is added by the implementation.
- 5910
- 5911 4. The collection of impressions is placed, in sequence, onto sides of the media sheets. This placement  
5912 is controlled by the "sides" attribute and the orientation of the print-stream page, as described in  
5913 section 4.2.8. The orientation of the print-stream pages affects the orientation of the impression;  
5914 for example, if "number-up" equals 2, then, typically, two portrait print-stream pages become one  
5915 landscape impression. Note that the placement of impressions onto media sheets is also controlled  
5916 by the "multiple-document-handling" attribute as described in section 4.2.4.
- 5917
- 5918 5. The "copies" and "multiple-document-handling" attributes are used to determine how many copies  
5919 of each media instance are created and in what order. See sections 4.2.5 and 4.2.4 for the details.  
5920

5921 6. When the correct number of copies are created, the media instances are finished according to the  
5922 values of the "finishings" attribute as described in 4.2.6. Note that sometimes finishing operations  
5923 may require manual intervention to perform the finishing operations on the copies, especially  
5924 uncollated copies. This specification allows any or all of the processing steps to be performed  
5925 automatically or manually at the discretion of the Printer object.

## 5926 17. APPENDIX E: Generic Directory Schema

5927 This section defines a generic schema for an entry in a directory service. A directory service is a means  
5928 by which service users can locate service providers. In IPP environments, this means that IPP Printers  
5929 can be registered (either automatically or with the help of an administrator) as entries of type printer in  
5930 the directory using an implementation specific mechanism such as entry attributes, entry type fields,  
5931 specific branches, etc. IPP clients can search or browse for entries of type printer. Clients use the  
5932 directory service to find entries based on naming, organizational contexts, or filtered searches on attribute  
5933 values of entries. For example, a client can find all printers in the "Local Department" context.  
5934 Authentication and authorization are also often part of a directory service so that an administrator can  
5935 place limits on end users so that they are only allowed to find entries to which they have certain access  
5936 rights. IPP itself does not require any specific directory service protocol or provider.

5937 Note: Some directory implementations allow for the notion of "aliasing". That is, one directory entry  
5938 object can appear as multiple directory entry object with different names for each object. In each case,  
5939 each alias refers to the same directory entry object which refers to a single IPP Printer object.

5940 The generic schema is a subset of IPP Printer Job Template and Printer Description attributes (sections  
5941 4.2 and 4.4). These attributes are identified as either **MANDATORYREQUIRED** or OPTIONAL for the  
5942 directory entry itself. This conformance labeling is NOT the same conformance labeling applied to the  
5943 attributes of IPP Printers objects. **MANDATORYREQUIRED** attributes MUST be associated with each  
5944 directory entry. OPTIONAL attributes SHOULD be associated with the directory entry (if known or  
5945 supported). In addition, all directory entry attributes SHOULD reflect the current attribute values for the  
5946 corresponding Printer object.

5947 In order to bridge between the directory service and the IPP Printer object, one of the  
5948 **MANDATORYREQUIRED** directory entry attributes is the Printer object's "printer-uri-supported"  
5949 attribute. The IPP client queries the "printer-uri-supported" attribute in the directory entry and then  
5950 addresses the IPP Printer object using one of its URIs. The "uri-security-supported" attribute identifies  
5951 the protocol (if any) used to secure a channel.

5952 The following attributes define the generic schema for directory entries of type PRINTER:

5953 printer-uri-supported **MANDATORYREQUIRED** Section 4.4.1



5954	uri-security-supported	<del>MANDATORYREQUIRED</del>	Section 4.4.2
5955	printer-name	<del>MANDATORYREQUIRED</del>	Section 4.4.3
5956	printer-location	OPTIONAL	Section 4.4.4
5957	printer-info	OPTIONAL	Section 4.4.5
5958	printer-more-info	OPTIONAL	Section 4.4.6
5959	printer-make-and-model	OPTIONAL	Section 4.4.8
5960	charset-supported	<del>MANDATORYREQUIRED</del>	Section 4.4.15
5961	generated-natural-language-		
5962	supported	<del>MANDATORYREQUIRED</del>	Section 4.4.17
5963	document-format-supported	OPTIONAL	Section 4.4.19
5964	color-supported	OPTIONAL	Section 4.4.23
5965	finishings-supported	OPTIONAL	Section 4.2.6
5966	number-up-supported	OPTIONAL	Section 4.2.7
5967	sides-supported	OPTIONAL	Section 4.2.8
5968	media-supported	OPTIONAL	Section 4.2.11
5969	printer-resolution-supported	OPTIONAL	Section 4.2.12
5970	print-quality-supported	OPTIONAL	Section 4.2.13
5971			