

1 INTERNET-DRAFT  
2 draft-ietf-ipp-model-v11-03.txt

R. deBry  
Utah Valley State College  
T. Hastings (editor)  
Xerox Corporation  
R. Herriot  
Xerox Corporation  
S. Isaacson  
Novell, Inc.  
P. Powell  
Astart Technologies  
June 11, 1998

15 Internet Printing Protocol/1.1: Model and Semantics  
16 Copyright (C) The Internet Society (1999). All Rights Reserved.

17 Status of this Memo

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

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

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

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

27 Abstract

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

37 The full set of IPP documents includes:

38 Design Goals for an Internet Printing Protocol [RFC2567]

39 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [RFC2568]

40 Internet Printing Protocol/1.1: Model and Semantics (this document)  
41 Internet Printing Protocol/1.1: Encoding and Transport [IPP-PRO]  
42 Internet Printing Protocol/1.1: Implementer's Guide [IPP-IIG]  
43 Mapping between LPD and IPP Protocols [RFC2569]  
44

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

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

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

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

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

67

**Table of Contents**

68	1. Introduction .....	10
69	1.1 Simplified Printing Model .....	11
70	2. IPP Objects .....	13
71	2.1 Printer Object .....	13
72	2.2 Job Object .....	15
73	2.3 Object Relationships .....	16
74	2.4 Object Identity .....	17
75	3. IPP Operations .....	19
76	3.1 Common Semantics .....	20
77	3.1.1 Required Parameters .....	20
78	3.1.2 Operation IDs and Request IDs .....	21
79	3.1.3 Attributes .....	21
80	3.1.4 Character Set and Natural Language Operation Attributes .....	22
81	3.1.4.1 Request Operation Attributes .....	23
82	3.1.4.2 Response Operation Attributes .....	25
83	3.1.5 Operation Targets .....	26
84	3.1.6 Operation Response Status Codes and Status Messages .....	28
85	3.1.6.1 "status-code" (type2 enum) .....	28
86	3.1.6.2 "status-message" (text(255)) .....	28
87	3.1.6.3 "detailed-status-message" (text(MAX)) .....	29
88	3.1.6.4 "document-access-error" (text(MAX)) .....	29
89	3.1.7 Unsupported Attributes .....	29
90	3.1.8 Versions .....	31
91	3.1.9 Job Creation Operations .....	32
92	3.2 Printer Operations .....	34
93	3.2.1 Print-Job Operation .....	34
94	3.2.1.1 Print-Job Request .....	34
95	3.2.1.2 Print-Job Response .....	38
96	3.2.2 Print-URI Operation .....	40
97	3.2.3 Validate-Job Operation .....	41
98	3.2.4 Create-Job Operation .....	41
99	3.2.5 Get-Printer-Attributes Operation .....	42
100	3.2.5.1 Get-Printer-Attributes Request .....	42
101	3.2.5.2 Get-Printer-Attributes Response .....	44
102	3.2.6 Get-Jobs Operation .....	45
103	3.2.6.1 Get-Jobs Request .....	45
104	3.2.6.2 Get-Jobs Response .....	46
105	3.2.7 Pause-Printer Operation .....	47
106	3.2.7.1 Pause-Printer Request .....	48
107	3.2.7.2 Pause-Printer Response .....	49
108	3.2.8 Resume-Printer Operation .....	49

109	3.2.9	Purge-Jobs Operation .....	50
110	3.3	Job Operations .....	51
111	3.3.1	Send-Document Operation .....	51
112	3.3.1.1	Send-Document Request .....	52
113	3.3.1.2	Send-Document Response .....	53
114	3.3.2	Send-URI Operation .....	53
115	3.3.3	Cancel-Job Operation .....	54
116	3.3.3.1	Cancel-Job Request .....	54
117	3.3.3.2	Cancel-Job Response .....	55
118	3.3.4	Get-Job-Attributes Operation .....	56
119	3.3.4.1	Get-Job-Attributes Request .....	56
120	3.3.4.2	Get-Job-Attributes Response .....	57
121	3.3.5	Hold-Job Operation .....	57
122	3.3.5.1	Hold-Job Request .....	58
123	3.3.5.2	Hold-Job Response .....	59
124	3.3.6	Release-Job Operation .....	59
125	3.3.7	Restart-Job Operation .....	60
126	3.3.7.1	Restart-Job Request .....	61
127	3.3.7.2	Restart-Job Response .....	62
128	4.	Object Attributes .....	62
129	4.1	Attribute Syntaxes .....	62
130	4.1.1	'text' .....	63
131	4.1.1.1	'textWithoutLanguage' .....	64
132	4.1.1.2	'textWithLanguage' .....	64
133	4.1.2	'name' .....	65
134	4.1.2.1	'nameWithoutLanguage' .....	65
135	4.1.2.2	'nameWithLanguage' .....	65
136	4.1.2.3	Matching 'name' attribute values .....	66
137	4.1.3	'keyword' .....	66
138	4.1.4	'enum' .....	67
139	4.1.5	'uri' .....	67
140	4.1.6	'uriScheme' .....	68
141	4.1.7	'charset' .....	68
142	4.1.8	'naturalLanguage' .....	69
143	4.1.9	'mimeType' .....	69
144	4.1.9.1	Application/octet-stream -- Auto-Sensing the document format .....	69
145	4.1.10	'octetString' .....	70
146	4.1.11	'boolean' .....	70
147	4.1.12	'integer' .....	71
148	4.1.13	'rangeOfInteger' .....	71
149	4.1.14	'dateTime' .....	71
150	4.1.15	'resolution' .....	71
151	4.1.16	'1setOf X' .....	71
152	4.2	Job Template Attributes .....	72

153	4.2.1	job-priority (integer(1:100)).....	75
154	4.2.2	job-hold-until (type3 keyword   name (MAX)).....	76
155	4.2.3	job-sheets (type3 keyword   name(MAX)).....	76
156	4.2.4	multiple-document-handling (type2 keyword).....	77
157	4.2.5	copies (integer(1:MAX)).....	78
158	4.2.6	finishings (1setOf type2 enum).....	78
159	4.2.7	page-ranges (1setOf rangeOfInteger (1:MAX)).....	80
160	4.2.8	sides (type2 keyword).....	81
161	4.2.9	number-up (integer(1:MAX)).....	81
162	4.2.10	orientation-requested (type2 enum).....	82
163	4.2.11	media (type3 keyword   name(MAX)).....	83
164	4.2.12	printer-resolution (resolution).....	83
165	4.2.13	print-quality (type2 enum).....	83
166	4.3	Job Description Attributes.....	84
167	4.3.1	job-uri (uri).....	86
168	4.3.2	job-id (integer(1:MAX)).....	86
169	4.3.3	job-printer-uri (uri).....	86
170	4.3.4	job-more-info (uri).....	87
171	4.3.5	job-name (name(MAX)).....	87
172	4.3.6	job-originating-user-name (name(MAX)).....	87
173	4.3.7	job-state (type1 enum).....	87
174	4.3.7.1	Forwarding Servers.....	89
175	4.3.7.2	Partitioning of Job States.....	90
176	4.3.8	job-state-reasons (1setOf type2 keyword).....	90
177	4.3.9	job-state-message (text(MAX)).....	93
178	4.3.10	job-detailed-status-messages (1setOf text(MAX)).....	94
179	4.3.11	job-document-access-errors (1setOf text(MAX)).....	94
180	4.3.12	number-of-documents (integer(0:MAX)).....	94
181	4.3.13	output-device-assigned (name(127)).....	94
182	4.3.14	Event Time Job Description Attributes.....	94
183	4.3.14.1	time-at-creation (integer(MIN:MAX)).....	95
184	4.3.14.2	time-at-processing (integer(MIN:MAX)).....	95
185	4.3.14.3	time-at-completed (integer(MIN:MAX)).....	95
186	4.3.14.4	job-printer-up-time (integer(1:MAX)).....	95
187	4.3.14.5	date-time-at-creation (dateTime).....	96
188	4.3.14.6	date-time-at-processing (dateTime).....	96
189	4.3.14.7	date-time-at-completed (dateTime).....	96
190	4.3.15	number-of-intervening-jobs (integer(0:MAX)).....	96
191	4.3.16	job-message-from-operator (text(127)).....	96
192	4.3.17	Job Size Attributes.....	96
193	4.3.17.1	job-k-octets (integer(0:MAX)).....	97
194	4.3.17.2	job-impressions (integer(0:MAX)).....	97
195	4.3.17.3	job-media-sheets (integer(0:MAX)).....	97
196	4.3.18	Job Progress Attributes.....	98
197	4.3.18.1	job-k-octets-processed (integer(0:MAX)).....	98

198	4.3.18.2	job-impressions-completed (integer(0:MAX)).....	98
199	4.3.18.3	job-media-sheets-completed (integer(0:MAX)) .....	98
200	4.3.19	attributes-charset (charset) .....	98
201	4.3.20	attributes-natural-language (naturalLanguage).....	98
202	4.4	Printer Description Attributes.....	99
203	4.4.1	printer-uri-supported (1setOf uri).....	101
204	4.4.2	uri-authentication-supported (1setOf type2 keyword) .....	101
205	4.4.3	uri-security-supported (1setOf type2 keyword).....	102
206	4.4.4	printer-name (name(127)) .....	103
207	4.4.5	printer-location (text(127)).....	103
208	4.4.6	printer-info (text(127)) .....	104
209	4.4.7	printer-more-info (uri).....	104
210	4.4.8	printer-driver-installer (uri) .....	104
211	4.4.9	printer-make-and-model (text(127)).....	104
212	4.4.10	printer-more-info-manufacturer (uri) .....	104
213	4.4.11	printer-state (type1 enum) .....	104
214	4.4.12	printer-state-reasons (1setOf type2 keyword) .....	105
215	4.4.13	printer-state-message (text(MAX)) .....	107
216	4.4.14	ipp-versions-supported (1setOf type2 keyword) .....	107
217	4.4.15	operations-supported (1setOf type2 enum) .....	107
218	4.4.16	multiple-document-jobs-supported (boolean) .....	108
219	4.4.17	charset-configured (charset) .....	109
220	4.4.18	charset-supported (1setOf charset).....	109
221	4.4.19	natural-language-configured (naturalLanguage) .....	109
222	4.4.20	generated-natural-language-supported (1setOf naturalLanguage) .....	109
223	4.4.21	document-format-default (mimeMediaType).....	110
224	4.4.22	document-format-supported (1setOf mimeMediaType) .....	110
225	4.4.23	printer-is-accepting-jobs (boolean) .....	110
226	4.4.24	queued-job-count (integer(0:MAX)).....	110
227	4.4.25	printer-message-from-operator (text(127)) .....	110
228	4.4.26	color-supported (boolean) .....	110
229	4.4.27	reference-uri-schemes-supported (1setOf uriScheme) .....	111
230	4.4.28	pdl-override-supported (type2 keyword).....	111
231	4.4.29	printer-up-time (integer(1:MAX)).....	111
232	4.4.30	printer-current-time (dateTime) .....	112
233	4.4.31	multiple-operation-time-out (integer(1:MAX)) .....	112
234	4.4.32	compression-supported (1setOf type3 keyword).....	112
235	4.4.33	job-k-octets-supported (rangeOfInteger(0:MAX)).....	113
236	4.4.34	job-impressions-supported (rangeOfInteger(0:MAX)) .....	113
237	4.4.35	job-media-sheets-supported (rangeOfInteger(0:MAX)).....	113
238	4.4.36	pages-per-minute (integer(0:MAX)) .....	113
239	4.4.37	pages-per-minute-color (integer(0:MAX)).....	113
240	5.	Conformance .....	114
241	5.1	Client Conformance Requirements .....	114

242	5.2	IPP Object Conformance Requirements.....	115
243	5.2.1	Objects.....	115
244	5.2.2	Operations .....	115
245	5.2.3	IPP Object Attributes .....	116
246	5.2.4	Versions.....	116
247	5.2.5	Extensions .....	117
248	5.2.6	Attribute Syntaxes .....	117
249	5.2.7	Security .....	117
250	5.3	Charset and Natural Language Requirements .....	117
251	6.	IANA Considerations (registered and private extensions).....	118
252	6.1	Typed 'keyword' and 'enum' Extensions .....	118
253	6.2	Attribute Extensibility .....	120
254	6.3	Attribute Syntax Extensibility .....	121
255	6.4	Operation Extensibility.....	121
256	6.5	Attribute Groups.....	121
257	6.6	Status Code Extensibility .....	122
258	6.7	Registration of MIME types/sub-types for document-formats.....	122
259	6.8	Registration of charsets for use in 'charset' attribute values .....	122
260	7.	Internationalization Considerations.....	122
261	8.	Security Considerations.....	125
262	8.1	Security Scenarios .....	126
263	8.1.1	Client and Server in the Same Security Domain.....	126
264	8.1.2	Client and Server in Different Security Domains.....	126
265	8.1.3	Print by Reference .....	127
266	8.2	URIs in Operation, Job, and Printer attributes .....	127
267	8.3	URIs for each authentication mechanisms .....	127
268	8.4	Restricted Queries .....	128
269	8.5	Operations performed by operators and system administrators.....	128
270	8.6	Queries on jobs submitted using non-IPP protocols.....	128
271	9.	References .....	129
272	10.	Author's Address.....	132
273	11.	Formats for IPP Registration Proposals .....	135
274	11.1	Type2 keyword attribute values registration .....	135
275	11.2	Type3 keyword attribute values registration .....	135
276	11.3	Type2 enum attribute values registration .....	135
277	11.4	Type3 enum attribute values registration .....	136
278	11.5	Attribute registration .....	136
279	11.6	Attribute Syntax registration .....	137
280	11.7	Operation registration.....	137
281	11.8	Attribute Group registration .....	137

282	11.9	Status code registration.....	138
283	12.	APPENDIX A: Terminology .....	138
284	12.1	Conformance Terminology.....	138
285	12.1.1	NEED NOT .....	138
286	12.2	Model Terminology.....	138
287	12.2.1	Keyword .....	138
288	12.2.2	Attributes.....	139
289	12.2.2.1	Attribute Name .....	139
290	12.2.2.2	Attribute Group Name.....	139
291	12.2.2.3	Attribute Value .....	139
292	12.2.2.4	Attribute Syntax.....	139
293	12.2.3	Supports.....	139
294	12.2.4	print-stream page.....	141
295	12.2.5	impression .....	141
296	13.	APPENDIX B: Status Codes and Suggested Status Code Messages .....	141
297	13.1	Status Codes .....	142
298	13.1.1	Informational .....	142
299	13.1.2	Successful Status Codes.....	142
300	13.1.2.1	successful-ok (0x0000).....	142
301	13.1.2.2	successful-ok-ignored-or-substituted-attributes (0x0001).....	143
302	13.1.2.3	successful-ok-conflicting-attributes (0x0002).....	143
303	13.1.3	Redirection Status Codes .....	143
304	13.1.4	Client Error Status Codes.....	143
305	13.1.4.1	client-error-bad-request (0x0400).....	143
306	13.1.4.2	client-error-forbidden (0x0401).....	144
307	13.1.4.3	client-error-not-authenticated (0x0402) .....	144
308	13.1.4.4	client-error-not-authorized (0x0403).....	144
309	13.1.4.5	client-error-not-possible (0x0404).....	144
310	13.1.4.6	client-error-timeout (0x0405).....	144
311	13.1.4.7	client-error-not-found (0x0406) .....	144
312	13.1.4.8	client-error-gone (0x0407) .....	145
313	13.1.4.9	client-error-request-entity-too-large (0x0408).....	145
314	13.1.4.10	client-error-request-value-too-long (0x0409).....	145
315	13.1.4.11	client-error-document-format-not-supported (0x040A) .....	146
316	13.1.4.12	client-error-attributes-or-values-not-supported (0x040B) .....	146
317	13.1.4.13	client-error-uri-scheme-not-supported (0x040C) .....	146
318	13.1.4.14	client-error-charset-not-supported (0x040D) .....	146
319	13.1.4.15	client-error-conflicting-attributes (0x040E) .....	146
320	13.1.4.16	client-error-compression-not-supported (0x040F) .....	147
321	13.1.4.17	client-error-compression-error (0x0410) .....	147
322	13.1.4.18	client-error-document-format-error (0x0411) .....	147
323	13.1.4.19	client-error-document-access-error (0x0412) .....	147
324	13.1.5	Server Error Status Codes .....	147
325	13.1.5.1	server-error-internal-error (0x0500) .....	147



326	13.1.5.2	server-error-operation-not-supported (0x0501).....	148
327	13.1.5.3	server-error-service-unavailable (0x0502) .....	148
328	13.1.5.4	server-error-version-not-supported (0x0503) .....	148
329	13.1.5.5	server-error-device-error (0x0504).....	148
330	13.1.5.6	server-error-temporary-error (0x0505) .....	149
331	13.1.5.7	server-error-not-accepting-jobs (0x0506).....	149
332	13.1.5.8	server-error-busy (0x0507).....	149
333	13.1.5.9	server-error-job-canceled (0x0508).....	149
334	13.1.5.10	server-error-multiple-document-jobs-not-supported (0x0509) .....	149
335	13.2	Status Codes for IPP Operations .....	150
336	14.	APPENDIX C: "media" keyword values .....	152
337	15.	APPENDIX D: Processing IPP Attributes .....	155
338	15.1	Fidelity.....	156
339	15.2	Page Description Language (PDL) Override.....	157
340	15.3	Using Job Template Attributes During Document Processing.....	158
341	16.	APPENDIX E: Generic Directory Schema.....	159
342	17.	APPENDIX F: Differences between the IPP/1.0 and IPP/1.1 "Model and Semantics" Documents	162
343	18.	Full Copyright Statement .....	166
344			

## 345 1. Introduction

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

350 Design Goals for an Internet Printing Protocol [RFC2567]  
351 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [RFC2568]  
352 Internet Printing Protocol/1.1: Model and Semantics (this document)  
353 Internet Printing Protocol/1.1: Encoding and Transport [IPP-PRO]  
354 Internet Printing Protocol/1.1: Implementer's Guide [IPP-IIG]  
355 Mapping between LPD and IPP Protocols [RFC2569]  
356

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

359 This document is laid out as follows:

- 360 - The rest of Section 1 is an introduction to the IPP simplified model for distributed printing.
- 361 - Section 2 introduces the object types covered in the model with their basic behaviors, attributes, and  
362 interactions.
- 363 - Section 3 defines the operations included in IPP/1.1. IPP operations are synchronous, therefore, for  
364 each operation, there is a both request and a response.
- 365 - Section 4 defines the attributes (and their syntaxes) that are used in the model.
- 366 - Sections 5 - 6 summarizes the implementation conformance requirements for objects that support the  
367 protocol and IANA considerations, respectively.
- 368 - Sections 7 - 11 cover the Internationalization and Security considerations as well as References,  
369 Author contact information, and Formats for Registration Proposals.
- 370 - Sections 12 - 14 are appendices that cover Terminology, Status Codes and Messages, and "media"  
371 keyword values.

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

- 378 - Section 15 is an appendix that helps to clarify the effects of interactions between related attributes and  
379 their values.
- 380 - Section 16 is an appendix that enumerates the subset of Printer attributes that form a generic directory  
381 schema. These attributes are useful when registering a Printer so that a client can find the Printer  
382 not just by name, but by filtered searches as well.
- 383 - Section 17 is an appendix summarizing the additions and changes from the IPP/1.0 "Model and  
384 Semantics" document [RFC2566] to make this IPP/1.1 document.
- 385 - Section 18 is the full copyright notice.

## 386 1.1 Simplified Printing Model

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

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

- 402 - Printer (Section 2.1)
- 403 - Job (Section 2.2)

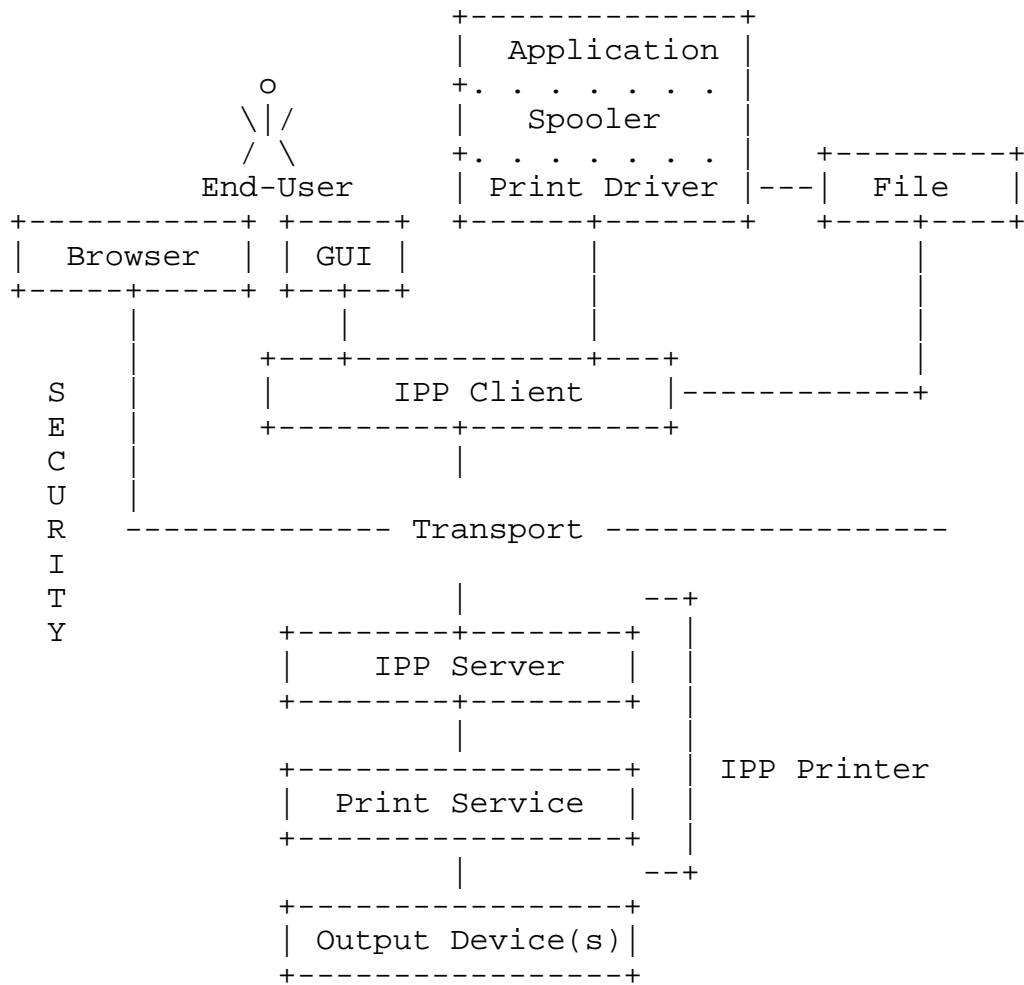
404

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

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

410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440

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



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

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

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

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

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

## 471 2. IPP Objects

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

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

- 481 - "REQUIRED": each object MUST support the attribute.
  - 482 - "RECOMMENDED": each object SHOULD support the attribute.
  - 483 - "OPTIONAL": each object MAY support the attribute.
- 484

485 Some definitions of attribute values indicate that an object MUST or SHOULD support the value;  
486 otherwise, support of the value is OPTIONAL. However, if an implementation supports an attribute, it  
487 MUST support at least one of the possible values for that attribute.

### 488 2.1 Printer Object

489 The major component of the IPP/1.1 model is the Printer object. A Printer object implements the server-  
490 side of the IPP/1.1 protocol. Using the protocol, end users may query the attributes of the Printer object and  
491 submit print jobs to the Printer object. The actual implementation components behind the Printer

492 abstraction may take on different forms and different configurations. However, the model abstraction  
493 allows the details of the configuration of real components to remain opaque to the end user. Section 3  
494 describes each of the Printer operations in detail.

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

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

501

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

505 Some examples of configurations supporting a Printer object include:

- 506 1) An output device with no spooling capabilities
- 507 2) An output device with a built-in spooler
- 508 3) A print server supporting IPP with one or more associated output devices
  - 509 3a) The associated output devices may or may not be capable of spooling jobs
  - 510 3b) The associated output devices may or may not support IPP

511

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

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

- 517 1. contained within software controlled by an end user, e.g. activated by the "Print" menu item in an  
518 application or
- 519 2. the print server component that sends IPP requests to either an output device or another  
520 "downstream" print server.

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

- 523 1. an (embedded) device component that accepts IPP requests and controls the device or
- 524 2. a component of a print server that accepts IPP requests (where the print server controls one or more  
525 networked devices using IPP or other protocols).

526

527 Legend:

528

529 ##### indicates a Printer object which is  
530 either embedded in an output device or is  
531 hosted in a server. The Printer object  
532 might or might not be capable of queuing/spooling.

533

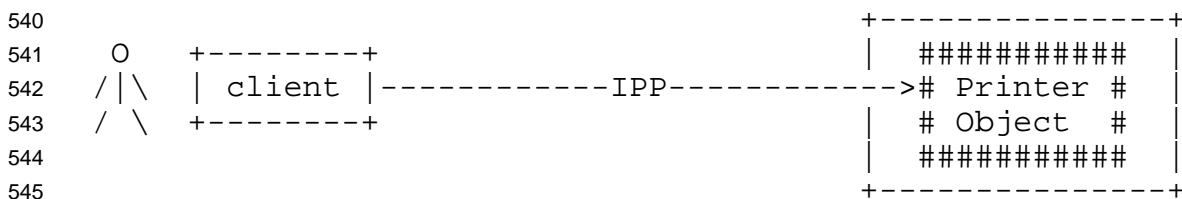
534 any indicates any network protocol or direct  
535 connect, including IPP

536

537

538 embedded printer:

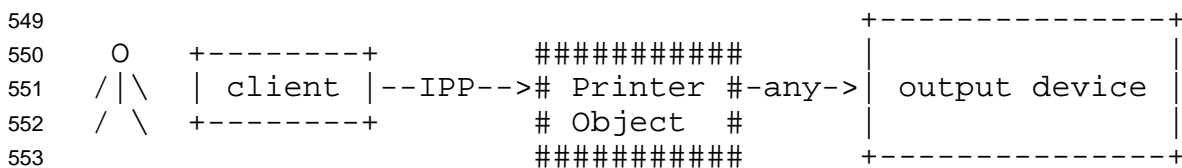
539



546

547

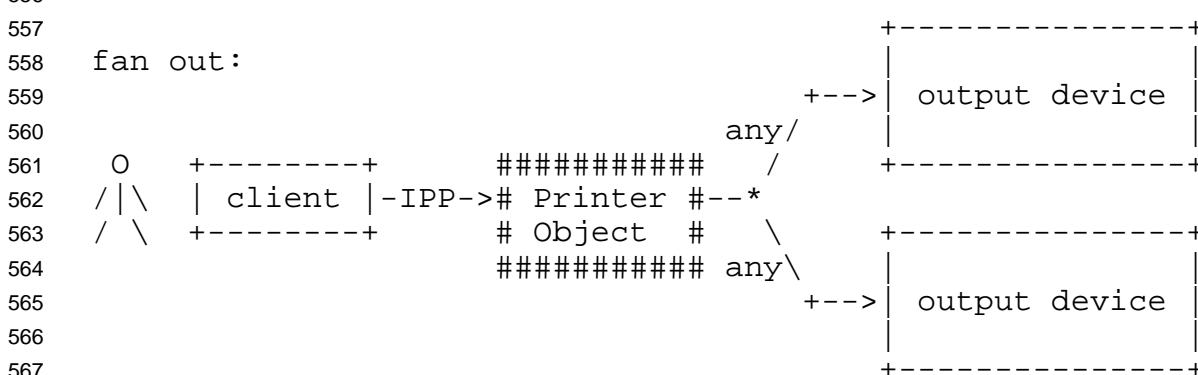
548 hosted printer:



554

555

556



567

568

569

## 570 2.2 Job Object

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

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

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

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

582

583 An implementation MUST support at least one document per Job object. An implementation MAY support  
584 multiple documents per Job object. A document is either:

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

588

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

## 592 2.3 Object Relationships

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

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

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

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



## 608 2.4 Object Identity

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

615 An administrator configures Printer objects to either support or not support authentication and/or message  
616 privacy using Transport Layer Security (TLS) [RFC2246] (the mechanism for security configuration is  
617 outside the scope of this IPP/1.1 document). In some situations, both types of connections (both  
618 authenticated and unauthenticated) can be established using a single communication channel that has some  
619 sort of negotiation mechanism. In other situations, multiple communication channels are used, one for each  
620 type of security configuration. Section 8 provides a full description of all security considerations and  
621 configurations.

622 If a Printer object supports more than one communication channel, some or all of those channels might  
623 support and/or require different security mechanisms. In such cases, an administrator could expose the  
624 simultaneous support for these multiple communication channels as multiple URIs for a single Printer  
625 object where each URI represents one of the communication channels to the Printer object. To support this  
626 flexibility, the IPP Printer object type defines a multi-valued identification attribute called the "printer-uri-  
627 supported" attribute. It MUST contain at least one URI. It MAY contain more than one URI. That is,  
628 every Printer object will have at least one URI that identifies at least one communication channel to the  
629 Printer object, but it may have more than one URI where each URI identifies a different communication  
630 channel to the Printer object. The "printer-uri-supported" attribute has two companion attributes, the "uri-  
631 security-supported" attribute and the "uri-authentication-supported". Both have the same cardinality as  
632 "printer-uri-supported". The purpose of the "uri-security-supported" attribute is to indicate the security  
633 mechanisms (if any) used for each URI listed in "printer-uri-supported". The purpose of the "uri-  
634 authentication-supported" attribute is to indicate the authentication mechanisms (if any) used for each URI  
635 listed in "printer-uri-supported". These three attributes are fully described in sections 4.4.1, 4.4.2, and  
636 4.4.3.

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

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

646 When a client submits a create request to the Printer object, the Printer object validates the request and  
647 creates a new Job object. The Printer object assigns the new Job object a URI which is stored in the "job-

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

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

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

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

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

686 To summarize:

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

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

### 710 3. IPP Operations

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

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

723 The IPP/1.1 Printer operations are:

- 724 Print-Job (section 3.2.1)
- 725 Print-URI (section 3.2.2)
- 726 Validate-Job (section 3.2.3)
- 727 Create-Job (section 3.2.4)
- 728 Get-Printer-Attributes (section 3.2.5)

729 Get-Jobs (section 3.2.6)  
730 Pause-Printer (section 3.3.5)  
731 Resume-Printer (section 3.3.6)  
732 Purge-Jobs (section 3.3.7)  
733

734 The Job operations are:

735 Send-Document (section 3.3.1)  
736 Send-URI (section 3.3.2)  
737 Cancel-Job (section 3.3.3)  
738 Get-Job-Attributes (section 3.3.4)  
739 Hold-Job (section 3.3.5)  
740 Release-Job (section 3.3.6)  
741 Restart-Job (section 3.3.7)  
742

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

### 745 3.1 Common Semantics

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

#### 748 3.1.1 Required Parameters

749 Every operation request contains the following REQUIRED parameters:

- 750 - a "version-number",
  - 751 - an "operation-id",
  - 752 - a "request-id", and
  - 753 - the attributes that are REQUIRED for that type of request.
- 754

755 Every operation response contains the following REQUIRED parameters:

- 756 - a "version-number",
  - 757 - a "status-code",
  - 758 - the "request-id" that was supplied in the corresponding request, and
  - 759 - the attributes that are REQUIRED for that type of response.
- 760

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

## 764 3.1.2 Operation IDs and Request IDs

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

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

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

## 780 3.1.3 Attributes

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

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

804 - Unsupported Attributes: In a create request, the client supplies a set of Operation and Job Template  
805 attributes. If any of these attributes or their values is unsupported by the Printer object, the Printer  
806 object returns the set of unsupported attributes in the response. Sections 3.1.7, 3.2.1.2, and 15 give  
807 a full description of how Job Template attributes supplied by the client in a create request are  
808 processed by the Printer object and how unsupported attributes are returned to the client. Because  
809 of extensibility, any IPP object might receive a request that contains new or unknown attributes or  
810 values for which it has no support. In such cases, the IPP object processes what it can and returns  
811 the unsupported attributes in the response. The Unsupported Attribute group is defined for all  
812 operation responses for returning unsupported attributes that the client supplied in the request.  
813

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

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

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

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

### 833 3.1.4 Character Set and Natural Language Operation Attributes

834 Some Job and Printer attributes have values that are text strings and names intended for human  
835 understanding rather than machine understanding (see the 'text' and 'name' attribute syntax descriptions in  
836 section 4.1). The following sections describe two special Operation Attributes called "attributes-charset"  
837 and "attributes-natural-language". These attributes are always part of the Operation Attributes group. For  
838 most attribute groups, the order of the attributes within the group is not important. However, for these two  
839 attributes within the Operation Attributes group, the order is critical. The "attributes-charset" attribute  
840 MUST be the first attribute in the group and the "attributes-natural-language" attribute MUST be the second  
841 attribute in the group. In other words, these attributes MUST be supplied in every IPP request and  
842 response, they MUST come first in the group, and MUST come in the specified order. For job creation  
843 operations, the IPP Printer implementation saves these two attributes with the new Job object as Job

844 Description attributes. For the sake of brevity in this document, these operation attribute descriptions are  
845 not repeated with every operation request and response, but have a reference back to this section instead.

#### 846 3.1.4.1 Request Operation Attributes

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

849 "attributes-charset" (charset):

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

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

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

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

877  
878 "attributes-natural-language" (naturalLanguage):

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

883  
884 There are no **REQUIRED** natural languages required for the Printer object to support. However, the  
885 Printer object's "generated-natural-language-supported" attribute identifies the natural languages  
886 supported by the Printer object and any contained Job objects for all text strings generated by the

887 IPP object. A client MAY query this attribute to determine which natural language(s) are supported  
888 for generated messages.

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

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

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

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

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

- 924
- 925 – In the implicit case, the value contains only the text/name value, and the language is  
926 specified by the "attributes-natural-language" operation attribute in the request or response  
927 (see sections 4.1.1.1 textWithoutLanguage and 4.1.2.1 nameWithoutLanguage).
  - 928
  - 929 – In the explicit case (also known as the Natural-Language Override case), the value contains  
930 both the language and the text/name value (see sections 4.1.1.2 textWithLanguage and  
931 4.1.2.2 nameWithLanguage).



932

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

943

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

946

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

949

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

956 - In a create request: If the client supplies a text or name attribute (for example, the "job-name"  
957 operation attribute) that uses an apparently incompatible combination, it is a client choice that does  
958 not affect the Printer object or its correct operation. Therefore, the Printer object simply accepts the  
959 client supplied value, stores it with the Job object, and responds back with the same combination  
960 whenever the client (or any client) queries for that attribute.

961 -In a query-type operation, like Get-Printer-Attributes: If the client requests an apparently incompatible  
962 combination, the Printer object responds (as described in section 3.1.4.2) using the Printer's  
963 configured natural language rather than the natural language requested by the client.

964

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

### 970 3.1.4.2 Response Operation Attributes

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

973 "attributes-charset" (charset):

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

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

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

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

996 This operation attribute identifies the natural language used by any 'text' and 'name' attributes that  
997 the IPP object is returning in this response. Unlike the "attributes-charset" operation attribute, the  
998 IPP object **NEED NOT** return the same value as that supplied by the client in the request. The IPP  
999 object **MAY** return the natural language of the Job object or the Printer's configured natural  
1000 language as identified by the Printer object's "natural-language-configured" attribute, rather than the  
1001 natural language supplied by the client. For any 'text' or 'name' attribute or status message in the  
1002 response that is in a different natural language than the value returned in the "attributes-natural-  
1003 language" operation attribute, the IPP object **MUST** use the Natural Language Override mechanism  
1004 (see sections 4.1.1.2 and 4.1.2.2) on each attribute value returned. The IPP object **MAY** use the  
1005 Natural Language Override mechanism redundantly, i.e., use it even when the value is in the same  
1006 natural language as the value supplied in the "attributes-natural-language" operation attribute of the  
1007 response.

### 1008 3.1.5 Operation Targets

1009 All IPP operations are directed at IPP objects. For Printer operations, the operation is always directed at a  
1010 Printer object using one of its URIs (i.e., one of the values in the Printer object's "printer-uri-supported"  
1011 attribute). Even if the Printer object supports more than one URI, the client supplies only one URI as the  
1012 target of the operation. The client identifies the target object by supplying the correct URI in the "printer-  
1013 uri (uri)" operation attribute.

1014 For Job operations, the operation is directed at either:

- 1015 - The Job object itself using the Job object's URI. In this case, the client identifies the target object by  
1016 supplying the correct URI in the "job-uri (uri)" operation attribute.  
1017 - The Printer object that created the Job object using both the Printer objects URI and the Job object's  
1018 Job ID. Since the Printer object that created the Job object generated the Job ID, it MUST be able to  
1019 correctly associate the client supplied Job ID with the correct Job object. The client supplies the  
1020 Printer object's URI in the "printer-uri (uri)" operation attribute and the Job object's Job ID in the  
1021 "job-id (integer(1:MAX))" operation attribute.  
1022

1023 If the operation is directed at the Job object directly using the Job object's URI, the client MUST NOT  
1024 include the redundant "job-id" operation attribute.

1025 The operation target attributes are REQUIRED operation attributes that MUST be included in every  
1026 operation request. Like the charset and natural language attributes (see section 3.1.4), the operation target  
1027 attributes are specially ordered operation attributes. In all cases, the operation target attributes immediately  
1028 follow the "attributes-charset" and "attributes-natural-language" attributes within the operation attribute  
1029 group, however the specific ordering rules are:

- 1030 - In the case where there is only one operation target attribute (i.e., either only the "printer-uri" attribute  
1031 or only the "job-uri" attribute), that attribute MUST be the third attribute in the operation attributes  
1032 group.  
1033 - In the case where Job operations use two operation target attributes (i.e., the "printer-uri" and "job-id"  
1034 attributes), the "printer-uri" attribute MUST be the third attribute and the "job-id" attribute MUST  
1035 be the fourth attribute.  
1036

1037 In all cases, the target URIs contained within the body of IPP operation requests and responses must be in  
1038 absolute format rather than relative format (a relative URL identifies a resource with the scope of the HTTP  
1039 server, but does not include scheme, host or port).

1040 The following rules apply to the use of port numbers in URIs that identify IPP objects:

- 1041 1. If the URI scheme allows the port number to be explicitly included in the URI string, and a port  
1042 number is specified within the URI, then that port number MUST be used by the client to contact  
1043 the IPP object.  
1044
- 1045 2. If the URI scheme allows the port number to be explicitly included in the URI string, and a port  
1046 number is not specified within the URI, then default port number implied by that URI scheme  
1047 MUST be used by the client to contact the IPP object.  
1048
- 1049 3. If the URI scheme does not allow an explicit port number to be specified within the URI, then the  
1050 default port number implied by that URI MUST be used by the client to contact the IPP object.  
1051

1052 Note: The IPP "Encoding and Transport document [IPP-PRO] shows a mapping of IPP onto HTTP/1.1 and  
1053 defines a new default port number for using IPP over HTTP/1.1.

## 1054 3.1.6 Operation Response Status Codes and Status Messages

1055 Every operation response includes a REQUIRED "status-code" parameter and an OPTIONAL "status-  
 1056 message" operation attribute, and an OPTIONAL "detailed-status-message" operation attribute. The Print-  
 1057 URI and Send-URI response MAY include an OPTIONAL "document-access-error" operation attribute.

## 1058 3.1.6.1 "status-code" (type2 enum)

1059 The REQUIRED "status-code" parameter provides information on the processing of a request.

1060 The status code is intended for use by automata. A client implementation of IPP SHOULD convert status  
 1061 code values into any localized message that has semantic meaning to the end user.

1062 The "status-code" value is a numeric value that has semantic meaning. The "status-code" syntax is similar  
 1063 to a "type2 enum" (see section 4.1 on "Attribute Syntaxes") except that values can range only from 0x0000  
 1064 to 0x7FFF. Section 13 describes the status codes, assigns the numeric values, and suggests a corresponding  
 1065 status message for each status code for use by the client when the user's natural language is English.

1066 If the Printer performs an operation with no errors and it encounters no problems, it MUST return the status  
 1067 code 'successful-ok' in the response. See section 13.

1068 If the client supplies unsupported values for the following parameters or Operation attributes, the Printer  
 1069 object MUST reject the operation, NEED NOT return the unsupported attribute value in the Unsupported  
 1070 Attributes group, and MUST return the indicated status code:

Parameter/Attribute	Status code
version-number	server-error-version-not-supported
operation-id	server-error-operation-not-supported
attributes-charset	client-error-charset-not-supported
compression	client-error-compression-not-supported
document-format	client-error-document-format-not-supported
document-uri	client-error-uri-scheme-not-supported, client-error-document-access-error

1071

1072 If the client supplies unsupported values for other attributes, or unsupported attributes, the Printer returns  
 1073 the status code defined in section 3.1.7 on Unsupported Attributes.

## 1074 3.1.6.2 "status-message" (text(255))

1075 The OPTIONAL "status-message" operation attribute provides a short textual description of the status of  
 1076 the operation. The "status-message" attribute's syntax is "text(255)", so the maximum length is 255 octets  
 1077 (see section 4.1.1). The status message is intended for the human end user. If a response does include a  
 1078 "status-message" attribute, an IPP client NEED NOT examine or display the messages, however it  
 1079 SHOULD do so in some implementation specific manner. The "status-message" is especially useful for a  
 1080 later version of a Printer object to return as supplemental information for the human user to accompany a  
 1081 status code that an earlier version of a client might not understand.

1082 If the Printer object supports the "status-message" operation attribute, the Printer object MUST be able to  
1083 generate this message in any of the natural languages identified by the Printer object's "generated-natural-  
1084 language-supported" attribute (see the "attributes-natural-language" operation attribute specified in section  
1085 3.1.4.1. Section 13 suggests the text for the status message returned by the Printer for use with the English  
1086 natural language.

1087 As described in section 3.1.4.1 for any returned 'text' attribute, if there is a choice for generating this  
1088 message, the Printer object uses the natural language indicated by the value of the "attributes-natural-  
1089 language" in the client request if supported, otherwise the Printer object uses the value in the Printer  
1090 object's own "natural-language-configured" attribute.

1091 If the Printer object supports the "status-message" operation attribute, it SHOULD use the REQUIRED 'utf-  
1092 8' charset to return a status message for the following error status codes (see section 13): 'client-error-bad-  
1093 request', 'client-error-charset-not-supported', 'server-error-internal-error', 'server-error-operation-not-  
1094 supported', and 'server-error-version-not-supported'. In this case, it MUST set the value of the "attributes-  
1095 charset" operation attribute to 'utf-8' in the error response.

### 1096 3.1.6.3 "detailed-status-message" (text(MAX))

1097 The OPTIONAL "detailed-status-message" operation attribute provides additional more detailed technical  
1098 and implementation-specific information about the operation. The "detailed-status-message" attribute's  
1099 syntax is "text(MAX)", so the maximum length is 1023 octets (see section 4.1.1). If the Printer objects  
1100 supports the "detailed-status-message" operation attribute, neither the Printer nor the client localizes the  
1101 message, since it is intended for use by the system administrator or other experienced technical persons.  
1102 Clients MUST NOT attempt to parse the value of this attribute. See the "document-access-error" operation  
1103 attribute (section 3.1.6.4) for additional errors that a program can process.

### 1104 3.1.6.4 "document-access-error" (text(MAX))

1105 This OPTIONAL operation attribute provides additional information about any document access errors  
1106 encountered by the Printer before it returned a response to the Print-URI (section 3.2.2) or Send-URI  
1107 (section 3.3.1) operation. For errors in the protocol identified by the URI scheme in the "document-uri"  
1108 operation attribute, such as 'http:' or 'ftp:', the error code is returned in parentheses, followed by the URI.  
1109 For example:

```
1110     (404) http://ftp.pwg.org/pub/pwg/ipp/new_MOD/ipp-model-v11-990510.pdf  
1111
```

1112 Most Internet protocols use decimal error codes (unlike IPP), so the ASCII error code representation is in  
1113 decimal.

### 1114 3.1.7 Unsupported Attributes

1115 The Unsupported Attributes group contains attributes that are not supported by the operation. This group is  
1116 primarily for the job creation operations, but all operations can return this group.

1117 A Printer object **MUST** include an Unsupported Attributes group in a response if the status code is one of  
1118 the following: 'successful-ok-ignored-or-substituted-attributes', 'successful-ok-conflicting-attributes', 'client-  
1119 error-attributes-or-values-not-supported' or 'client-error-conflicting-attributes'.

1120 If the status code is one of the four specified in the preceding paragraph, the Unsupported Attributes group  
1121 **MUST** contain all of those attributes and only those attributes that are:

- 1122 a. an Operation or Job Template attribute supplied in the request, and
- 1123 b. unsupported by the printer. See below for details on the three categories "unsupported" attributes.

1124 If the Printer object is not returning any Unsupported Attributes in the response, the Printer object  
1125 **SHOULD** omit Group 2 rather than sending an empty group. However, a client **MUST** be able to accept an  
1126 empty group.

1127 Unsupported attributes fall into three categories:

- 1128 1. The Printer object does not support the supplied attribute (no matter what the attribute syntax or  
1129 value).
- 1130 2. The Printer object does support the attribute, but does not support some or all of the particular  
1131 attribute syntaxes or values supplied by the client (i.e., the Printer object does not have those  
1132 attribute syntaxes or values in its corresponding "xxx-supported" attribute).
- 1133 3. The Printer object does support the attributes and values supplied, but the particular values are in  
1134 conflict with one another, because they violate a constraint, such as not being able to staple  
1135 transparencies.

1136 In the case of an unsupported attribute name, the Printer object returns the client-supplied attribute with a  
1137 substituted value of 'unsupported'. This value's syntax type is "out-of-band" and its encoding is defined by  
1138 special rules for "out-of-band" values in the "Encoding and Transport" document [IPP-PRO]. Its value  
1139 indicates no support for the attribute itself (see the beginning of section 4.1).

1140 In the case of a supported attribute with one or more unsupported attribute syntaxes or values, the Printer  
1141 object simply returns the client-supplied attribute with the unsupported attribute syntaxes or values as  
1142 supplied by the client. This indicates support for the attribute, but no support for that particular attribute  
1143 syntax or value. If the client supplies a multi-valued attribute with more than one value and the Printer  
1144 object supports the attribute but only supports a subset of the client-supplied attribute syntaxes or values,  
1145 the Printer object **MUST** return only those attribute syntaxes or values that are unsupported.

1146 In the case of two (or more) supported attribute values that are in conflict with one another (although each  
1147 is supported independently, the values conflict when requested together within the same job), the Printer  
1148 object **MUST** return all the values that it ignores or substitutes to resolve the conflict, but not any of the  
1149 values that it is still using. The choice for exactly how to resolve the conflict is implementation dependent.  
1150 See sections 3.2.1.2 and 15. See The Implementer's Guide [IPP-IIG] for an example.

## 1151 3.1.8 Versions

1152 Each operation request and response carries with it a "version-number" parameter. Each value of the  
1153 "version-number" is in the form "X.Y" where X is the major version number and Y is the minor version  
1154 number. By including a version number in the client request, it allows the client to identify which version  
1155 of IPP it is interested in using, i.e., the version whose conformance requirements the client may be  
1156 depending upon the Printer to meet.

1157 If the IPP object does not support that major version number supplied by the client, i.e., the major version  
1158 field of the "version-number" parameter does not match any of the values of the Printer's "ipp-versions-  
1159 supported" (see section 4.4.14), the object MUST respond with a status code of 'server-error-version-not-  
1160 supported' along with the closest version number that is supported (see section 13.1.5.4). If the major  
1161 version number is supported, but the minor version number is not, the IPP object SHOULD accept and  
1162 attempt to perform the request (or reject the request if the operation is not supported), else it rejects the  
1163 request and returns the 'server-error-version-not-supported' status code. In all cases, the IPP object MUST  
1164 return the "version-number" that it supports that is closest to the version number supplied by the client in  
1165 the request.

1166 There is no version negotiation per se. However, if after receiving a 'server-error-version-not-supported'  
1167 status code from an IPP object, a client SHOULD try again with a different version number. A client MAY  
1168 also determine the versions supported either from a directory that conforms to Appendix E (see section 16)  
1169 or by querying the Printer object's "ipp-versions-supported" attribute (see section 4.4.14) to determine  
1170 which versions are supported.

1171 An IPP object implementation MUST support version '1.1', i.e., meet the conformance requirements for  
1172 IPP/1.1 as specified in this document and [IPP-PRO]. An IPP object implementation SHOULD support  
1173 version '1.0', i.e., meet the conformance requirements for IPP/1.0 [RFC2566 and RFC2565].

1174 There is only one notion of "version number" that covers both IPP Model and IPP Protocol changes. Thus  
1175 the version number MUST change when introducing a new version of the Model and Semantics document  
1176 (this document) or a new version of the "Encoding and Transport" document [IPP-PRO].

1177 Changes to the major version number of the Model and Semantics document indicate structural or syntactic  
1178 changes that make it impossible for older version of IPP clients and Printer objects to correctly parse and  
1179 correctly process the new or changed attributes, operations and responses. If the major version number  
1180 changes, the minor version numbers is set to zero. As an example, adding the REQUIRED "ipp-attribute-  
1181 fidelity" attribute to version '1.1' (if it had not been part of version '1.0'), would have required a change to  
1182 the major version number, since an IPP/1.0 Printer would not have processed a request with the correct  
1183 semantics that contained the "ipp-attribute-fidelity" attribute that it did not know about. Items that might  
1184 affect the changing of the major version number include any changes to the Model and Semantics document  
1185 (this document) or the "Encoding and Transport" document [IPP-PRO] itself, such as:

- 1186 - reordering of ordered attributes or attribute sets
- 1187 - changes to the syntax of existing attributes
- 1188 - adding REQUIRED (for an IPP object to support) operation attribute groups
- 1189 - adding values to existing REQUIRED operation attributes
- 1190 - adding REQUIRED operations

1191

1192 Changes to the minor version number indicate the addition of new features, attributes and attribute values  
1193 that may not be understood by all IPP objects, but which can be ignored if not understood. Items that might  
1194 affect the changing of the minor version number include any changes to the model objects and attributes but  
1195 not the encoding and transport rules [IPP-PRO] (except adding attribute syntaxes). Examples of such  
1196 changes are:

- 1197 - grouping all extensions not included in a previous version into a new version
- 1198 - adding new attribute values
- 1199 - adding new object attributes
- 1200 - adding OPTIONAL (for an IPP object to support) operation attributes (i.e., those attributes that an IPP  
1201 object can ignore without confusing clients)
- 1202 - adding OPTIONAL (for an IPP object to support) operation attribute groups (i.e., those attributes that  
1203 an IPP object can ignore without confusing clients)
- 1204 - adding new attribute syntaxes
- 1205 - adding OPTIONAL operations
- 1206 - changing Job Description attributes or Printer Description attributes from OPTIONAL to REQUIRED  
1207 or vice versa.
- 1208 - adding OPTIONAL attribute syntaxes to an existing attribute.

1209 The encoding of the "version-number" MUST NOT change over any version number (either major or  
1210 minor). This rule guarantees that all future versions will be backwards compatible with all previous  
1211 versions (at least for checking the "version-number"). In addition, any protocol elements (attributes, error  
1212 codes, tags, etc.) that are not carried forward from one version to the next are deprecated so that they can  
1213 never be reused with new semantics.

1214 Implementations that support a certain version NEED NOT support ALL previous versions. As each new  
1215 version is defined (through the release of a new IPP specification document), that version will specify  
1216 which previous versions MUST and which versions SHOULD be supported in compliant implementations.

### 1217 3.1.9 Job Creation Operations

1218 In order to "submit a print job" and create a new Job object, a client issues a create request. A create  
1219 request is any one of following three operation requests:

- 1220 - The Print-Job Request: A client that wants to submit a print job with only a single document uses the  
1221 Print-Job operation. The operation allows for the client to "push" the document data to the Printer  
1222 object by including the document data in the request itself.
- 1223
- 1224 - The Print-URI Request: A client that wants to submit a print job with only a single document (where  
1225 the Printer object "pulls" the document data instead of the client "pushing" the data to the Printer  
1226 object) uses the Print-URI operation. In this case, the client includes in the request only a URI  
1227 reference to the document data (not the document data itself).
- 1228
- 1229 - The Create-Job Request: A client that wants to submit a print job with multiple documents uses the  
1230 Create-Job operation. This operation is followed by an arbitrary number of Send-Document and/or  
1231 Send-URI operations (each creating another document for the newly create Job object). The Send-



1232 Document operation includes the document data in the request (the client "pushes" the document  
1233 data to the printer), and the Send-URI operation includes only a URI reference to the document data  
1234 in the request (the Printer "pulls" the document data from the referenced location). The last Send-  
1235 Document or Send-URI request for a given Job object includes a "last-document" operation attribute  
1236 set to 'true' indicating that this is the last request.  
1237

1238 Throughout this model document, the term "create request" is used to refer to any of these three operation  
1239 requests.

1240 A Create-Job operation followed by only one Send-Document operation is semantically equivalent to a  
1241 Print-Job operation, however, for performance reasons, the client SHOULD use the Print-Job operation for  
1242 all single document jobs. Also, Print-Job is a REQUIRED operation (all implementations MUST support  
1243 it) whereas Create-Job is an OPTIONAL operation, hence some implementations might not support it.

1244 Job submission time is the point in time when a client issues a create request. The initial state of every Job  
1245 object is the 'pending', 'pending-held', or 'processing' state (see section 4.3.7). When the Printer object  
1246 begins processing the print job, the Job object's state moves to 'processing'. This is known as job  
1247 processing time. There are validation checks that must be done at job submission time and others that must  
1248 be performed at job processing time.

1249 At job submission time and at the time a Validate-Job operation is received, the Printer MUST do the  
1250 following:

- 1251 1. Process the client supplied attributes and either accept or reject the request
  - 1252 2. Validate the syntax of and support for the scheme of any client supplied URI
- 1253

1254 At job submission time the Printer object MUST validate whether or not the supplied attributes, attribute  
1255 syntaxes, and values are supported by matching them with the Printer object's corresponding "xxx-  
1256 supported" attributes. See section 3.1.7 for details. [IPP-IIG] presents suggested steps for an IPP object to  
1257 either accept or reject any request and additional steps for processing create requests.

1258 At job submission time the Printer object NEED NOT perform the validation checks reserved for job  
1259 processing time such as:

- 1260 1. Validating the document data
  - 1261 2. Validating the actual contents of any client supplied URI (resolve the reference and follow the link to  
1262 the document data)
- 1263

1264 At job submission time, these additional job processing time validation checks are essentially useless, since  
1265 they require actually parsing and interpreting the document data, are not guaranteed to be 100% accurate,  
1266 and MUST be done, yet again, at job processing time. Also, in the case of a URI, checking for availability  
1267 at job submission time does not guarantee availability at job processing time. In addition, at job processing  
1268 time, the Printer object might discover any of the following conditions that were not detectable at job  
1269 submission time:

- 1270 - runtime errors in the document data,

- 1271 - nested document data that is in an unsupported format,  
1272 - the URI reference is no longer valid (i.e., the server hosting the document might be down), or  
1273 - any other job processing error  
1274

1275 At job submission time, a Printer object, especially a non-spooling Printer, MAY accept jobs that it does  
1276 not have enough space for. In such a situation, a Printer object MAY stop reading data from a client for an  
1277 indefinite period of time. A client MUST be prepared for a write operation to block for an indefinite period  
1278 of time (see section 5.1 on client conformance).

1279 When a Printer object has too little space for starting a new job, it MAY reject a new create request. In this  
1280 case, a Printer object MUST return a response (in reply to the rejected request) with a status-code of 'server-  
1281 error-busy' (see section 14.1.5.8) and it MAY close the connection before receiving all bytes of the  
1282 operation. A Printer SHOULD indicate that it is temporarily unable to accept jobs by setting the 'spool-  
1283 space-full' value in its "printer-state-reasons" attribute and removing the value when it can accept another  
1284 job (see section 4.4.12).

1285 When receiving a 'server-error-busy' status-code in an operation response, a client MUST be prepared for  
1286 the Printer object to close the connection before the client has sent all of the data (especially for the Print-  
1287 Job operation). A client MUST be prepared to keep submitting a create request until the IPP Printer object  
1288 accepts the create request.

1289 At job processing time, since the Printer object has already responded with a successful status code in the  
1290 response to the create request, if the Printer object detects an error, the Printer object is unable to inform the  
1291 end user of the error with an operation status code. In this case, the Printer, depending on the error, can set  
1292 the job object's "job-state", "job-state-reasons", or "job-state-message" attributes to the appropriate value(s)  
1293 so that later queries can report the correct job status.

1294 Note: Asynchronous notification of events is outside the scope of this IPP/1.1 document.

1295

## 1296 3.2 Printer Operations

1297 All Printer operations are directed at Printer objects. A client MUST always supply the "printer-uri"  
1298 operation attribute in order to identify the correct target of the operation.

### 1299 3.2.1 Print-Job Operation

1300 This REQUIRED operation allows a client to submit a print job with only one document and supply the  
1301 document data (rather than just a reference to the data). See Section 15 for the suggested steps for  
1302 processing create operations and their Operation and Job Template attributes.

#### 1303 3.2.1.1 Print-Job Request

1304 The following groups of attributes are supplied as part of the Print-Job Request:

## 1305 Group 1: Operation Attributes

## 1306 Natural Language and Character Set:

1307 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.1.  
1308 The Printer object MUST copy these values to the corresponding Job Description attributes  
1309 described in sections 4.3.19 and 4.3.20.

1310

## 1311 Target:

1312 The "printer-uri" (uri) operation attribute which is the target for this operation as described in  
1313 section 3.1.5.

1314

## 1315 Requesting User Name:

1316 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1317 described in section 8.3.

1318

## 1319 "job-name" (name(MAX)):

1320 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It  
1321 contains the client supplied Job name. If this attribute is supplied by the client, its value is used for  
1322 the "job-name" attribute of the newly created Job object. The client MAY automatically include any  
1323 information that will help the end-user distinguish amongst his/her jobs, such as the name of the  
1324 application program along with information from the document, such as the document name,  
1325 document subject, or source file name. If this attribute is not supplied by the client, the Printer  
1326 generates a name to use in the "job-name" attribute of the newly created Job object (see Section  
1327 4.3.5).

1328

## 1329 "ipp-attribute-fidelity" (boolean):

1330 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute.  
1331 The value 'true' indicates that total fidelity to client supplied Job Template attributes and values is  
1332 required, else the Printer object MUST reject the Print-Job request. The value 'false' indicates that a  
1333 reasonable attempt to print the Job object is acceptable and the Printer object MUST accept the  
1334 Print-job request. If not supplied, the Printer object assumes the value is 'false'. All Printer objects  
1335 MUST support both types of job processing. See section 15 for a full description of "ipp-attribute-  
1336 fidelity" and its relationship to other attributes, especially the Printer object's "pdl-override-  
1337 supported" attribute.

1338

## 1339 "document-name" (name(MAX)):

1340 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute.  
1341 It contains the client supplied document name. The document name MAY be different than the Job  
1342 name. Typically, the client software automatically supplies the document name on behalf of the end  
1343 user by using a file name or an application generated name. If this attribute is supplied, its value can  
1344 be used in a manner defined by each implementation. Examples include: printed along with the Job  
1345 (job start sheet, page adornments, etc.), used by accounting or resource tracking management tools,  
1346 or even stored along with the document as a document level attribute. IPP/1.1 does not support the  
1347 concept of document level attributes.

1348

1349 "compression" (type3 keyword)

1350 The client **OPTIONALLY** supplies this attribute. The Printer object **MUST** support this attribute  
1351 and the "compression-supported" attribute (see section 4.4.32). The client supplied "compression"  
1352 operation attribute identifies the compression algorithm used on the document data. The following  
1353 cases exist:

- 1354 a) If the client omits this attribute, the Printer object **MUST** assume that the data is not  
1355 compressed (i.e. the Printer follows the rules below as if the client supplied the  
1356 "compression" attribute with a value of 'none').
- 1357 b) If the client supplies this attribute, but the value is not supported by the Printer object,  
1358 i.e., the value is not one of the values of the Printer object's "compression-supported"  
1359 attribute, the Printer object **MUST** reject the request, and return the 'client-error-  
1360 compression-not-supported' status code. See section 3.1.7 for returning unsupported  
1361 attributes and values.
- 1362 c) If the client supplies the attribute and the Printer object supports the attribute value, the  
1363 Printer object uses the corresponding decompression algorithm on the document data.
- 1364 d) If the decompression algorithm fails before the Printer returns an operation response, the  
1365 Printer object **MUST** reject the request and return the 'client-error-compression-error'  
1366 status code.
- 1367 e) If the decompression algorithm fails after the Printer returns an operation response, the  
1368 Printer object **MUST** abort the job and add the 'compression-error' value to the job's  
1369 "job-state-reasons" attribute.
- 1370 f) If the decompression algorithm succeeds, the document data **MUST** then have the format  
1371 specified by the job's "document-format" attribute, if supplied (see "document-format"  
1372 operation attribute definition below).

1373 "document-format" (mimeMediaType) :

1374 The client **OPTIONALLY** supplies this attribute. The Printer object **MUST** support this attribute.  
1375 The value of this attribute identifies the format of the supplied document data. The following cases  
1376 exist:  
1377

- 1378 a) If the client does not supply this attribute, the Printer object assumes that the document  
1379 data is in the format defined by the Printer object's "document-format-default" attribute.  
1380 (i.e. the Printer follows the rules below as if the client supplied the "document-format"  
1381 attribute with a value equal to the printer's default value).
- 1382 b) If the client supplies this attribute, but the value is not supported by the Printer object,  
1383 i.e., the value is not one of the values of the Printer object's "document-format-  
1384 supported" attribute, the Printer object **MUST** reject the request and return the 'client-  
1385 error-document-format-not-supported' status code.
- 1386 c) If the client supplies this attribute and its value is 'application/octet-stream' (i.e. to be  
1387 auto-sensed, see Section 4.1.9.1), and the format is not one of the document-formats that  
1388 the Printer can auto-sense, and this check occurs before the Printer returns an operation  
1389 response, then the Printer **MUST** reject the request and return the 'client-error-  
1390 document-format-not-supported' status code.
- 1391 d) If the client supplies this attribute, and the value is supported by the Printer object, the  
1392 document data, the Printer is capable of interpreting the document data.

- 1393 e) If interpreting of the document data fails before the Printer returns an operation response,  
1394 the Printer object MUST reject the request and return the 'client-error-document-format-  
1395 error' status code.
- 1396 f) If interpreting of the document data fails after the Printer returns an operation response,  
1397 the Printer object MUST abort the job and add the 'document-format-error' value to the  
1398 job's "job-state-reasons" attribute.

1399  
1400 "document-natural-language" (naturalLanguage):

1401 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this  
1402 attribute. This attribute specifies the natural language of the document for those document-formats  
1403 that require a specification of the natural language in order to image the document unambiguously.  
1404 There are no particular values required for the Printer object to support.

1405  
1406  
1407 "job-k-octets" (integer(0:MAX))

1408 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this  
1409 attribute and the "job-k-octets-supported" attribute (see section 4.4.33). The client supplied "job-k-  
1410 octets" operation attribute identifies the total size of the document(s) in K octets being submitted  
1411 (see section 4.3.17.1 for the complete semantics). If the client supplies the attribute and the Printer  
1412 object supports the attribute, the value of the attribute is used to populate the Job object's "job-k-  
1413 octets" Job Description attribute.

1414  
1415 For this attribute and the following two attributes ("job-impressions", and "job-media-sheets"), if the  
1416 client supplies the attribute, but the Printer object does not support the attribute, the Printer object  
1417 ignores the client-supplied value. If the client supplies the attribute and the Printer supports the  
1418 attribute, and the value is within the range of the corresponding Printer object's "xxx-supported"  
1419 attribute, the Printer object MUST use the value to populate the Job object's "xxx" attribute. If the  
1420 client supplies the attribute and the Printer supports the attribute, but the value is outside the range  
1421 of the corresponding Printer object's "xxx-supported" attribute, the Printer object MUST copy the  
1422 attribute and its value to the Unsupported Attributes response group, reject the request, and return  
1423 the 'client-error-attributes-or-values-not-supported' status code. If the client does not supply the  
1424 attribute, the Printer object MAY choose to populate the corresponding Job object attribute  
1425 depending on whether the Printer object supports the attribute and is able to calculate or discern the  
1426 correct value.

1427  
1428 "job-impressions" (integer(0:MAX))

1429 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this  
1430 attribute and the "job-impressions-supported" attribute (see section 4.4.34). The client supplied  
1431 "job-impressions" operation attribute identifies the total size in number of impressions of the  
1432 document(s) being submitted (see section 4.3.17.2 for the complete semantics).

1433  
1434 See last paragraph under "job-k-octets".  
1435

1436 "job-media-sheets" (integer(0:MAX))

1437 The client **OPTIONALLY** supplies this attribute. The Printer object **OPTIONALLY** supports this  
1438 attribute and the "job-media-sheets-supported" attribute (see section 4.4.35). The client supplied  
1439 "job-media-sheets" operation attribute identifies the total number of media sheets to be produced for  
1440 this job (see section 4.3.17.3 for the complete semantics).

1441  
1442 See last paragraph under "job-k-octets".  
1443

#### 1444 Group 2: Job Template Attributes

1445 The client **OPTIONALLY** supplies a set of Job Template attributes as defined in section 4.2. If the  
1446 client is not supplying any Job Template attributes in the request, the client **SHOULD** omit Group 2  
1447 rather than sending an empty group. However, a Printer object **MUST** be able to accept an empty  
1448 group.  
1449

#### 1450 Group 3: Document Content

1451 The client **MUST** supply the document data to be processed.  
1452

1453 In addition to the **MANDATORY** parameters required for every operation request, the simplest Print-Job  
1454 Request consists of just the "attributes-charset" and "attributes-natural-language" operation attributes; the  
1455 "printer-uri" target operation attribute; the Document Content and nothing else. In this simple case, the  
1456 Printer object:

- 1457 - creates a new Job object (the Job object contains a single document),
- 1458 - stores a generated Job name in the "job-name" attribute in the natural language and charset requested  
1459 (see Section 3.1.4.1) (if those are supported, otherwise using the Printer object's default natural  
1460 language and charset), and
- 1461 - at job processing time, uses its corresponding default value attributes for the supported Job Template  
1462 attributes that were not supplied by the client as IPP attribute or embedded instructions in the  
1463 document data.  
1464

#### 1465 3.2.1.2 Print-Job Response

1466 The Printer object **MUST** return to the client the following sets of attributes as part of the Print-Job  
1467 Response:

##### 1468 Group 1: Operation Attributes

###### 1469 Status Message:

1470 In addition to the **REQUIRED** status code returned in every response, the response **OPTIONALLY**  
1471 includes a "status-message" (text(255)) operation attribute as described in sections 13 and 3.1.6. If  
1472 the client supplies unsupported or conflicting Job Template attributes or values, the Printer object  
1473 **MUST** reject or accept the Print-Job request depending on the whether the client supplied a 'true' or

1474 'false' value for the "ipp-attribute-fidelity" operation attribute. See the Implementer's Guide [IPP-  
1475 IIG] for a complete description of the suggested steps for processing a create request.

1476

1477 Natural Language and Character Set:

1478 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.2.

1479

1480 Group 2: Unsupported Attributes

1481 See section 3.1.7 for details on returning Unsupported Attributes.

1482

1483 The value of the "ipp-attribute-fidelity" supplied by the client does not affect what attributes the  
1484 Printer object returns in this group. The value of "ipp-attribute-fidelity" only affects whether the  
1485 Print-Job operation is accepted or rejected. If the job is accepted, the client may query the job using  
1486 the Get-Job-Attributes operation requesting the unsupported attributes that were returned in the  
1487 create response to see which attributes were ignored (not stored on the Job object) and which  
1488 attributes were stored with other (substituted) values.

1489

1490 Group 3: Job Object Attributes

1491 "job-uri" (uri):

1492 The Printer object MUST return the Job object's URI by returning the contents of the REQUIRED  
1493 "job-uri" Job object attribute. The client uses the Job object's URI when directing operations at the  
1494 Job object. The Printer object always uses its configured security policy when creating the new  
1495 URI. However, if the Printer object supports more than one URI, the Printer object also uses  
1496 information about which URI was used in the Print-Job Request to generate the new URI so that  
1497 the new URI references the correct access channel. In other words, if the Print-Job Request comes  
1498 in over a secure channel, the Printer object MUST generate a Job URI that uses the secure channel  
1499 as well.

1500

1501 "job-id" (integer(1:MAX)):

1502 The Printer object MUST return the Job object's Job ID by returning the REQUIRED "job-id" Job  
1503 object attribute. The client uses this "job-id" attribute in conjunction with the "printer-uri" attribute  
1504 used in the Print-Job Request when directing Job operations at the Printer object.

1505

1506 "job-state":

1507 The Printer object MUST return the Job object's REQUIRED "job-state" attribute. The value of this  
1508 attribute (along with the value of the next attribute: "job-state-reasons") is taken from a "snapshot"  
1509 of the new Job object at some meaningful point in time (implementation defined) between when the  
1510 Printer object receives the Print-Job Request and when the Printer object returns the response.

1511

1512 "job-state-reasons":

1513 The Printer object MUST return the Job object's REQUIRED "job-state-reasons" attribute.

1514

1515 "job-state-message":

1516 The Printer object OPTIONALLY returns the Job object's OPTIONAL "job-state-message"  
1517 attribute. If the Printer object supports this attribute then it MUST be returned in the response. If

1518 this attribute is not returned in the response, the client can assume that the "job-state-message"  
1519 attribute is not supported and will not be returned in a subsequent Job object query.

1520  
1521 "number-of-intervening-jobs":

1522 The Printer object OPTIONALLY returns the Job object's OPTIONAL "number-of-intervening-  
1523 jobs" attribute. If the Printer object supports this attribute then it MUST be returned in the response.  
1524 If this attribute is not returned in the response, the client can assume that the "number-of-  
1525 intervening-jobs" attribute is not supported and will not be returned in a subsequent Job object  
1526 query.

1527  
1528 Note: Since any printer state information which affects a job's state is reflected in the "job-state" and  
1529 "job-state-reasons" attributes, it is sufficient to return only these attributes and no specific printer  
1530 status attributes.

1531  
1532 Note: In addition to the MANDATORY parameters required for every operation response, the simplest  
1533 response consists of the just the "attributes-charset" and "attributes-natural-language" operation attributes  
1534 and the "job-uri", "job-id", and "job-state" Job Object Attributes. In this simplest case, the status code is  
1535 'successful-ok' and there is no "status-message" operation attribute.

### 1536 3.2.2 Print-URI Operation

1537 This OPTIONAL operation is identical to the Print-Job operation (section 3.2.1) except that a client  
1538 supplies a URI reference to the document data using the "document-uri" (uri) operation attribute (in Group  
1539 1) rather than including the document data itself. Before returning the response, the Printer MUST validate  
1540 that the Printer supports the retrieval method (e.g., http, ftp, etc.) implied by the URI, and MUST check for  
1541 valid URI syntax. If the client-supplied URI scheme is not supported, i.e. the value is not in the Printer  
1542 object's "referenced-uri-scheme-supported" attribute, the Printer object MUST reject the request and return  
1543 the 'client-error-uri-scheme-not-supported' status code.

1544 The IPP Printer MAY validate the accessibility of the document as part of the operation or subsequently. If  
1545 the Printer determines an accessibility problem before returning an operation response, it rejects the request  
1546 and returns the 'client-error-document-access-error' status code. The Printer MAY also return a specific  
1547 document access error code using the "document-access-error" operation attribute (see section 3.1.6.4).

1548 If the Printer determines this document accessibility problem after accepting the request and returning an  
1549 operation response with one of the successful status codes, the Printer adds the 'document-access-error'  
1550 value to the job's "job-state-reasons" attribute and MAY populate the job's "job-document-access-errors"  
1551 Job Description attribute (see section 4.3.11). See The Implementer's Guide [IPP-IIG] for suggested  
1552 additional checks.

1553 If the Printer object supports this operation, it MUST support the "reference-uri-schemes-supported" Printer  
1554 attribute (see section 4.4.27).

1555 It is up to the IPP object to interpret the URI and subsequently "pull" the document from the source  
1556 referenced by the URI string.



## 1557 3.2.3 Validate-Job Operation

1558 This REQUIRED operation is similar to the Print-Job operation (section 3.2.1) except that a client supplies  
1559 no document data and the Printer allocates no resources (i.e., it does not create a new Job object). This  
1560 operation is used only to verify capabilities of a printer object against whatever attributes are supplied by  
1561 the client in the Validate-Job request. By using the Validate-Job operation a client can validate that an  
1562 identical Print-Job operation (with the document data) would be accepted. The Validate-Job operation also  
1563 performs the same security negotiation as the Print-Job operation (see section 8), so that a client can check  
1564 that the client and Printer object security requirements can be met before performing a Print-Job operation.

1565 The Validate-Job operation does not accept a "document-uri" attribute in order to allow a client to check  
1566 that the same Print-URI operation will be accepted, since the client doesn't send the data with the Print-URI  
1567 operation. The client SHOULD just issue the Print-URI request.

1568 The Printer object returns the same status codes, Operation Attributes (Group 1) and Unsupported  
1569 Attributes (Group 2) as the Print-Job operation. However, no Job Object Attributes (Group 3) are returned,  
1570 since no Job object is created.

## 1571 3.2.4 Create-Job Operation

1572 This OPTIONAL operation is similar to the Print-Job operation (section 3.2.1) except that in the Create-Job  
1573 request, a client does not supply document data or any reference to document data. Also, the client does not  
1574 supply any of the "document-name", "document-format", "compression", or "document-natural-language"  
1575 operation attributes. This operation is followed by one or more Send-Document or Send-URI operations.  
1576 In each of those operation requests, the client OPTIONALLY supplies the "document-name", "document-  
1577 format", and "document-natural-language" attributes for each document in the multi-document Job object.

1578 If a Printer object supports the Create-Job operation, it MUST also support the Send-Document operation  
1579 and also MAY support the Send-URI operation.

1580 If the Printer object supports this operation, it MUST support the "multiple-operation-time-out" Printer  
1581 attribute (see section 4.4.31).

1582 If the Printer object supports this operation, then it MUST support the "multiple-document-jobs-supported"  
1583 Printer Description attribute (see section 4.4.16) and indicate whether or not it supports multiple-document  
1584 jobs.

1585 If the Printer object supports this operation and supports multiple documents in a job, then it MUST support  
1586 the "multiple-document-handling" Job Template job attribute with at least one value (see section 4.2.4) and  
1587 the associated "multiple-document-handling-default" and "multiple-document-handling-supported" Job  
1588 Template Printer attributes (see section 4.2).

1589 After the Create-Job operation has completed, the value of the "job-state" attribute is similar to the "job-  
1590 state" after a Print-Job, even though no document-data has arrived. A Printer MAY set the 'job-data-  
1591 insufficient' value of the job's "job-state-reason" attribute to indicate that processing cannot begin until  
1592 sufficient data has arrived and set the "job-state" to either 'pending' or 'pending-held'. A non-spooling

1593 printer that doesn't implement the 'pending' job state may even set the "job-state" to 'processing', even  
1594 though there is not yet any data to process. See sections 4.3.7 and 4.3.8.

### 1595 3.2.5 Get-Printer-Attributes Operation

1596 This REQUIRED operation allows a client to request the values of the attributes of a Printer object. In the  
1597 request, the client supplies the set of Printer attribute names and/or attribute group names in which the  
1598 requester is interested. In the response, the Printer object returns a corresponding attribute set with the  
1599 appropriate attribute values filled in.

1600 For Printer objects, the possible names of attribute groups are:

- 1601 - 'job-template': the subset of the Job Template attributes that apply to a Printer object (the last two  
1602 columns of the table in Section 4.2) that the implementation supports for Printer objects.
- 1603 - 'printer-description': the subset of the attributes specified in Section 4.4 that the implementation  
1604 supports for Printer objects.
- 1605 - 'all': the special group 'all' that includes all attributes that the implementation supports for Printer  
1606 objects.

1607

1608 Since a client MAY request specific attributes or named groups, there is a potential that there is some  
1609 overlap. For example, if a client requests, 'printer-name' and 'all', the client is actually requesting the  
1610 "printer-name" attribute twice: once by naming it explicitly, and once by inclusion in the 'all' group. In such  
1611 cases, the Printer object NEED NOT return each attribute only once in the response even if it is requested  
1612 multiple times. The client SHOULD NOT request the same attribute in multiple ways.

1613 It is NOT REQUIRED that a Printer object support all attributes belonging to a group (since some attributes  
1614 are OPTIONAL). However, it is REQUIRED that each Printer object support all group names.

#### 1615 3.2.5.1 Get-Printer-Attributes Request

1616 The following sets of attributes are part of the Get-Printer-Attributes Request:

##### 1617 Group 1: Operation Attributes

1618 Natural Language and Character Set:

1619 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.1.

1620

1621 Target:

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

1624

1625 Requesting User Name:

1626 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1627 described in section 8.3.

1628

1629 "requested-attributes" (1setOf keyword) :

1630 The client OPTIONALLY supplies a set of attribute names and/or attribute group names in whose  
1631 values the requester is interested. The Printer object MUST support this attribute. If the client  
1632 omits this attribute, the Printer MUST respond as if this attribute had been supplied with a value of  
1633 'all'.  
1634

1635 "document-format" (mimeMediaType) :

1636 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute.  
1637 This attribute is useful for a Printer object to determine the set of supported attribute values that  
1638 relate to the requested document format. The Printer object MUST return the attributes and values  
1639 that it uses to validate a job on a create or Validate-Job operation in which this document format is  
1640 supplied. The Printer object SHOULD return only (1) those attributes that are supported for the  
1641 specified format and (2) the attribute values that are supported for the specified document format.  
1642 By specifying the document format, the client can get the Printer object to eliminate the attributes  
1643 and values that are not supported for a specific document format. For example, a Printer object  
1644 might have multiple interpreters to support both 'application/postscript' (for PostScript) and  
1645 'text/plain' (for text) documents. However, for only one of those interpreters might the Printer  
1646 object be able to support "number-up" with values of '1', '2', and '4'. For the other interpreter it  
1647 might be able to only support "number-up" with a value of '1'. Thus a client can use the Get-Printer-  
1648 Attributes operation to obtain the attributes and values that will be used to accept/reject a create job  
1649 operation.  
1650

1651 If the Printer object does not distinguish between different sets of supported values for each  
1652 different document format when validating jobs in the create and Validate-Job operations, it MUST  
1653 NOT distinguish between different document formats in the Get-Printer-Attributes operation. If the  
1654 Printer object does distinguish between different sets of supported values for each different  
1655 document format specified by the client, this specialization applies only to the following Printer  
1656 object attributes:  
1657

- 1658 - Printer attributes that are Job Template attributes ("xxx-default" "xxx-supported", and "xxx-  
1659 ready" in the Table in Section 4.2),
  - 1660 - "pdl-override-supported",
  - 1661 - "compression-supported",
  - 1662 - "job-k-octets-supported",
  - 1663 - "job-impressions-supported",
  - 1664 - "job-media-sheets-supported"
  - 1665 - "printer-driver-installer",
  - 1666 - "color-supported", and
  - 1667 - "reference-uri-schemes-supported"
- 1668

1669 The values of all other Printer object attributes (including "document-format-supported") remain  
1670 invariant with respect to the client supplied document format (except for new Printer description  
1671 attribute as registered according to section 6.2).  
1672

1673 If the client omits this "document-format" operation attribute, the Printer object MUST respond as if  
1674 the attribute had been supplied with the value of the Printer object's "document-format-default"  
1675 attribute. It is recommended that the client always supply a value for "document-format", since the  
1676 Printer object's "document-format-default" may be 'application/octet-stream', in which case the  
1677 returned attributes and values are for the union of the document formats that the Printer can  
1678 automatically sense. For more details, see the description of the 'mimeType' attribute syntax  
1679 in section 4.1.9.

1680  
1681 If the client supplies a value for the "document-format" Operation attribute that is not supported by  
1682 the Printer, i.e., is not among the values of the Printer object's "document-format-supported"  
1683 attribute, the Printer object MUST reject the operation and return the 'client-error-document-format-  
1684 not-supported' status code.  
1685

### 1686 3.2.5.2 Get-Printer-Attributes Response

1687 The Printer object returns the following sets of attributes as part of the Get-Printer-Attributes Response:

#### 1688 Group 1: Operation Attributes

##### 1689 Status Message:

1690 In addition to the REQUIRED status code returned in every response, the response OPTIONALLY  
1691 includes a "status-message" (text(255)) operation attribute as described in sections 13 and 3.1.6.  
1692

##### 1693 Natural Language and Character Set:

1694 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.2.  
1695

#### 1696 Group 2: Unsupported Attributes

1697 See section 3.1.7 for details on returning Unsupported Attributes.  
1698

1699 The response NEED NOT contain the "requested-attributes" operation attribute with any supplied  
1700 values (attribute keywords) that were requested by the client but are not supported by the IPP object.  
1701 If the Printer object does include unsupported attributes referenced in "requested-attributes" and  
1702 such attributes include group names, such as 'all', the unsupported attributes MUST NOT include  
1703 attributes described in the standard but not supported by the implementation.  
1704

#### 1705 Group 3: Printer Object Attributes

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

1714

## 1715 3.2.6 Get-Jobs Operation

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

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

## 1722 3.2.6.1 Get-Jobs Request

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

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

## 1725 Group 1: Operation Attributes

## 1726 Natural Language and Character Set:

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

1728

## 1729 Target:

1730 The "printer-uri" (uri) operation attribute which is the target for this operation as described in  
1731 section 3.1.5.

1732

## 1733 Requesting User Name:

1734 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1735 described in section 8.3.

1736

## 1737 "limit" (integer(1:MAX)):

1738 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It  
1739 is an integer value that determines the maximum number of jobs that a client will receive from the  
1740 Printer even if "which-jobs" or "my-jobs" constrain which jobs are returned. The limit is a "stateless  
1741 limit" in that if the value supplied by the client is 'N', then only the first 'N' jobs are returned in the  
1742 Get-Jobs Response. There is no mechanism to allow for the next 'M' jobs after the first 'N' jobs. If  
1743 the client does not supply this attribute, the Printer object responds with all applicable jobs.

1744

## 1745 "requested-attributes" (1setOf keyword):

1746 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It  
1747 is a set of Job attribute names and/or attribute groups names in whose values the requester is  
1748 interested. This set of attributes is returned for each Job object that is returned. The allowed  
1749 attribute group names are the same as those defined in the Get-Job-Attributes operation in section  
1750 3.3.4. If the client does not supply this attribute, the Printer MUST respond as if the client had  
1751 supplied this attribute with two values: 'job-uri' and 'job-id'.

1752

"which-jobs" (keyword):

1753

1754

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

1755

1756

1757

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

1758

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

1759

1760

1761

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

1762

1763

1764

1765

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

1766

1767

1768

1769

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

1770

1771

1772

"my-jobs" (boolean):

1773

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

1774

1775

1776

1777

1778

1779

### 3.2.6.2 Get-Jobs Response

1780

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

1781

1782

1783

1784

1785

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

1786

1787

1788

### Group 1: Operation Attributes

1789

Status Message:

1790

In addition to the **REQUIRED** status code returned in every response, the response **OPTIONALLY** includes a "status-message" (text(255)) operation attribute as described in sections 13 and 3.1.6.

1791

1792

1793

1794 Natural Language and Character Set:

1795 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.2.

1796

1797 Group 2: Unsupported Attributes

1798 See section 3.1.7 for details on returning Unsupported Attributes.

1799

1800 The response NEED NOT contain the "requested-attributes" operation attribute with any supplied  
1801 values (attribute keywords) that were requested by the client but are not supported by the IPP object.1802 If the Printer object does include unsupported attributes referenced in "requested-attributes" and  
1803 such attributes include group names, such as 'all', the unsupported attributes MUST NOT include  
1804 attributes described in the standard but not supported by the implementation.

1805

1806 Groups 3 to N: Job Object Attributes

1807 The Printer object responds with one set of Job Object Attributes for each returned Job object. The  
1808 Printer object ignores (does not respond with) any requested attribute or value which is not  
1809 supported or which is restricted by the security policy in force, including whether the requesting  
1810 user is the user that submitted the job (job originating user) or not (see section 8). However, the  
1811 Printer object MUST respond with the 'unknown' value for any supported attribute (including all  
1812 REQUIRED attributes) for which the Printer object does not know the value, unless it would violate  
1813 the security policy. See the description of the "out-of-band" values in the beginning of Section 4.1.

1814

1815 Jobs are returned in the following order:

- 1816 - If the client requests all 'completed' Jobs (Jobs in the 'completed', 'aborted', or 'canceled' states),
- 
- 1817 then the Jobs are returned newest to oldest (with respect to actual completion time)
- 
- 1818 - If the client requests all 'not-completed' Jobs (Jobs in the 'pending', 'processing', 'pending-held',
- 
- 1819 and 'processing-stopped' states), then Jobs are returned in relative chronological order of
- 
- 1820 expected time to complete (based on whatever scheduling algorithm is configured for the
- 
- 1821 Printer object).

1822 3.2.7 Pause-Printer Operation

1823 This OPTIONAL operation allows a client to stop the Printer object from scheduling jobs on all its devices.  
1824 Depending on implementation, the Pause-Printer operation MAY also stop the Printer from processing the  
1825 current job or jobs. Any job that is currently being printed is either stopped as soon as the implementation  
1826 permits or is completed, depending on implementation. The Printer object MUST still accept create  
1827 operations to create new jobs, but MUST prevent any jobs from entering the 'processing' state.1828 If the Pause-Printer operation is supported, then the Resume-Printer operation MUST be supported, and  
1829 vice-versa.1830 The IPP Printer stops the current job(s) on its device(s) that were in the 'processing' or 'processing-stopped'  
1831 states as soon as the implementation permits. If the implementation will take appreciable time to stop, the

1832 IPP Printer adds the 'moving-to-paused' value to the Printer object's "printer-state-reasons" attribute (see  
 1833 section 4.4.12). When the device(s) have all stopped, the IPP Printer transitions the Printer object to the  
 1834 'stopped' state, removes the 'moving-to-paused' value, if present, and adds the 'paused' value to the Printer  
 1835 object's "printer-state-reasons" attribute.

1836 When the current job(s) complete that were in the 'processing' state, the IPP Printer transitions them to the  
 1837 'completed' state. When the current job(s) stop in mid processing that were in the 'processing' state, the IPP  
 1838 Printer transitions them to the 'processing-stopped' state and adds the 'printer-stopped' value to the job's  
 1839 "job-state-reasons" attribute.

1840 For any jobs that are 'pending' or 'pending-held', the 'printer-stopped' value of the jobs' "job-state-reasons"  
 1841 attribute also applies. However, the IPP Printer NEED NOT update those jobs' "job-state-reasons"  
 1842 attributes and only need return the 'printer-stopped' value when those jobs are queried (so-called "lazy  
 1843 evaluation").

1844 Whether the Pause-Printer operation affects jobs that were submitted to the device from other sources than  
 1845 the IPP Printer object in the same way that the Pause-Printer operation affects jobs that were submitted to  
 1846 the IPP Printer object using IPP, depends on implementation, i.e., on whether the IPP protocol is being used  
 1847 as a universal management protocol or just to manage IPP jobs, respectively.

1848 The IPP Printer MUST accept the request in any state and transition the Printer to the indicated new  
 1849 "printer-state" before returning as follows:

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

1850 *Access Rights:* The authenticated user (see section 8.3) performing this operation must be an operator or  
 1851 administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP Printer MUST reject the  
 1852 operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized'  
 1853 as appropriate.

### 1854 3.2.7.1 Pause-Printer Request

1855 The following groups of attributes are part of the Pause-Printer Request:

#### 1856 Group 1: Operation Attributes

1857 Natural Language and Character Set:

1858 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.1.



1859

**Target:**

1860

1861

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

1862

1863

1864

**Requesting User Name:**

1865

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

1866

1867

**3.2.7.2 Pause-Printer Response**

1868

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

1869

**Group 1: Operation Attributes**

1870

**Status Message:**

1871

In addition to the REQUIRED status code returned in every response, the response OPTIONALLY includes a "status-message" (text(255)) operation attribute as described in sections 13 and 3.1.6.

1872

1873

1874

**Natural Language and Character Set:**

1875

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

1876

1877

**Group 2: Unsupported Attributes**

1878

See section 3.1.7 for details on returning Unsupported Attributes.

1879

1880

**3.2.8 Resume-Printer Operation**

1881

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

1882

1883

1884

1885

1886

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

1887

1888

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

1889

1890

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

1891

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

1892 *Access Rights:* The authenticated user (see section 8.3) performing this operation must be an operator or  
 1893 administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP Printer MUST reject the  
 1894 operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized'  
 1895 as appropriate.

1896 The Resume-Printer Request and Resume-Printer Response have the same attribute groups and attributes as  
 1897 the Pause-Printer operation (see sections 3.2.7.1 and 3.2.7.2).

### 1898 3.2.9 Purge-Jobs Operation

1899 This OPTIONAL operation allows a client to remove all jobs from an IPP Printer object, regardless of their  
 1900 job states, including jobs in the Printer object's Job History (see Section 4.3.7.2). After a Purge-Jobs  
 1901 operation has been performed, a Printer object MUST return no jobs in subsequent Get-Job-Attributes and  
 1902 Get-Jobs responses (until new jobs are submitted).

1903 Whether the Purge-Jobs (and Get-Jobs) operation affects jobs that were submitted to the device from other  
 1904 sources than the IPP Printer object in the same way that the Purge-Jobs operation affects jobs that were  
 1905 submitted to the IPP Printer object using IPP, depends on implementation, i.e., on whether the IPP protocol  
 1906 is being used as a universal management protocol or just to manage IPP jobs, respectively.

1907 Note: if an operator wants to cancel all jobs without clearing out the Job History, the operator uses the  
 1908 Cancel-Job operation on each job instead of using the Purge-Job operation.

1909 The Printer object MUST accept this operation in any state and transition the Printer object to the 'idle'  
 1910 state.

1911 *Access Rights:* The authenticated user (see section 8.3) performing this operation must be an operator or  
 1912 administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP object MUST reject the  
 1913 operation and return: client-error-forbidden, client-error-not-authenticated, and client-error-not-authorized  
 1914 as appropriate.

1915 The Purge-Jobs Request and Purge-Jobs Response have the same attribute groups and attributes as the  
 1916 Pause-Printer operation (see sections 3.2.7.1 and 3.2.7.2).

1917

## 1918 3.3 Job Operations

1919 All Job operations are directed at Job objects. A client MUST always supply some means of identifying the  
1920 Job object in order to identify the correct target of the operation. That job identification MAY either be a  
1921 single Job URI or a combination of a Printer URI with a Job ID. The IPP object implementation MUST  
1922 support both forms of identification for every job.

## 1923 3.3.1 Send-Document Operation

1924 This OPTIONAL operation allows a client to create a multi-document Job object that is initially "empty"  
1925 (contains no documents). In the Create-Job response, the Printer object returns the Job object's URI (the  
1926 "job-uri" attribute) and the Job object's 32-bit identifier (the "job-id" attribute). For each new document  
1927 that the client desires to add, the client uses a Send-Document operation. Each Send-Document Request  
1928 contains the entire stream of document data for one document.

1929 If the Printer supports this operation but does not support multiple documents per job, the Printer MUST  
1930 reject subsequent Send-Document operations supplied with data and return the 'server-error-multiple-  
1931 document-jobs-not-supported'. However, the Printer MUST accept the first document with a 'true' or 'false'  
1932 value for the "last-document" operation attribute (see below), so that clients MAY always submit one  
1933 document jobs with a 'false' value for "last-document" in the first Send-Document and a 'true' for "last-  
1934 document" in the second Send-Document (with no data).

1935 Since the Create-Job and the send operations (Send-Document or Send-URI operations) that follow could  
1936 occur over an arbitrarily long period of time for a particular job, a client MUST send another send operation  
1937 within an IPP Printer defined minimum time interval after the receipt of the previous request for the job. If  
1938 a Printer object supports multiple document jobs, the Printer object MUST support the "multiple-operation-  
1939 time-out" attribute (see section 4.4.31). This attribute indicates the minimum number of seconds the Printer  
1940 object will wait for the next send operation before taking some recovery action.

1941 An IPP object MUST recover from an errant client that does not supply a send operation, sometime after  
1942 the minimum time interval specified by the Printer object's "multiple-operation-time-out" attribute. Such  
1943 recovery MAY include any of the following or other recovery actions:

- 1944 1. Assume that the Job is an invalid job, start the process of changing the job state to 'aborted', add the  
1945 'aborted-by-system' value to the job's "job-state-reasons" attribute (see section 4.3.8), and clean up  
1946 all resources associated with the Job. In this case, if another send operation is finally received, the  
1947 Printer responds with an "client-error-not-possible" or "client-error-not-found" depending on  
1948 whether or not the Job object is still around when the send operation finally arrives.
- 1949 2. Assume that the last send operation received was in fact the last document (as if the "last-document"  
1950 flag had been set to 'true'), close the Job object, and proceed to process it (i.e., move the Job's state  
1951 to 'pending').
- 1952 3. Assume that the last send operation received was in fact the last document, close the Job, but move it  
1953 to the 'pending-held' and add the 'submission-interrupted' value to the job's "job-state-reasons"  
1954 attribute (see section 4.3.8). This action allows the user or an operator to determine whether to  
1955 continue processing the Job by moving it back to the 'pending' state using the Release-Job operation  
1956 (see section 3.3.6) or to cancel the job using the Cancel-Job operation (see section 3.3.3).

1957

1958 Each implementation is free to decide the "best" action to take depending on local policy, whether any  
1959 documents have been added, whether the implementation spools jobs or not, and/or any other piece of  
1960 information available to it. If the choice is to abort the Job object, it is possible that the Job object may  
1961 already have been processed to the point that some media sheet pages have been printed.

1962 *Access Rights:* The authenticated user (see section 8.3) performing this operation must either be the job  
1963 owner (as determined in the Create-Job operation) or an operator or administrator of the Printer object (see  
1964 Sections 1 and 8.5). Otherwise, the IPP object MUST reject the operation and return: 'client-error-  
1965 forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' as appropriate.

### 1966 3.3.1.1 Send-Document Request

1967 The following attribute sets are part of the Send-Document Request:

#### 1968 Group 1: Operation Attributes

1969 Natural Language and Character Set:

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

1971

1972 Target:

1973 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation  
1974 attribute(s) which define the target for this operation as described in section 3.1.5.

1975

1976 Requesting User Name:

1977 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1978 described in section 8.3.

1979

1980 "document-name" (name(MAX)):

1981 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It  
1982 contains the client supplied document name. The document name MAY be different than the Job  
1983 name. It might be helpful, but NEED NOT be unique across multiple documents in the same Job.  
1984 Typically, the client software automatically supplies the document name on behalf of the end user  
1985 by using a file name or an application generated name. See the description of the "document-name"  
1986 operation attribute in the Print-Job Request (section 3.2.1.1) for more information about this  
1987 attribute.

1988

1989 "compression" (type3 keyword)

1990 See the description of "compression" for the Print-Job operation in Section 3.2.1.1.

1991

1992 "document-format" (mimeMediaType) :

1993 See the description of "document-format" for the Print-Job operation in Section 3.2.1.1.

1994

1995 "document-natural-language" (naturalLanguage):

1996 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this  
1997 attribute. This attribute specifies the natural language of the document for those document-formats

1998 that require a specification of the natural language in order to image the document unambiguously.  
1999 There are no particular values required for the Printer object to support.

2001  
2002 "last-document" (boolean):

2003 The client **MUST** supply this attribute. The Printer object **MUST** support this attribute. It is a  
2004 boolean flag that is set to 'true' if this is the last document for the Job, 'false' otherwise.  
2005

## 2006 Group 2: Document Content

2007 The client **MUST** supply the document data if the "last-document" flag is set to 'false'. However,  
2008 since a client might not know that the previous document sent with a Send-Document (or Send-  
2009 URI) operation was the last document (i.e., the "last-document" attribute was set to 'false'), it is legal  
2010 to send a Send-Document request with no document data where the "last-document" flag is set to  
2011 'true'. Such a request **MUST NOT** increment the value of the Job object's "number-of-documents"  
2012 attribute, since no real document was added to the job.

### 2013 3.3.1.2 Send-Document Response

2014 The following sets of attributes are part of the Send-Document Response:

#### 2015 Group 1: Operation Attributes

2016 Status Message:

2017 In addition to the **REQUIRED** status code returned in every response, the response **OPTIONALLY**  
2018 includes a "status-message" (text(255)) operation attribute as described in sections 13 and 3.1.6.  
2019

2020 Natural Language and Character Set:

2021 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.2.  
2022

#### 2023 Group 2: Unsupported Attributes

2024 See section 3.1.7 for details on returning Unsupported Attributes.

#### 2025 Group 3: Job Object Attributes

2026 This is the same set of attributes as described in the Print-Job response (see section 3.2.1.2).  
2027

### 2028 3.3.2 Send-URI Operation

2029 This **OPTIONAL** operation is identical to the Send-Document operation (see section 3.3.1) except that a  
2030 client **MUST** supply a URI reference ("document-uri" operation attribute) rather than the document data  
2031 itself. If a Printer object supports this operation, clients can use both Send-URI or Send-Document  
2032 operations to add new documents to an existing multi-document Job object. However, if a client needs to  
2033 indicate that the previous Send-URI or Send-Document was the last document, the client **MUST** use the  
2034 Send-Document operation with no document data and the "last-document" flag set to 'true' (rather than  
2035 using a Send-URI operation with no "document-uri" operation attribute).

2036 If a Printer object supports this operation, it **MUST** also support the Print-URI operation (see section 3.2.2).

2037 The Printer object **MUST** validate the syntax and URI scheme of the supplied URI before returning a  
 2038 response, just as in the Print-URI operation. The IPP Printer **MAY** validate the accessibility of the  
 2039 document as part of the operation or subsequently (see section 3.2.2).

### 2040 3.3.3 Cancel-Job Operation

2041 This **REQUIRED** operation allows a client to cancel a Print Job from the time the job is created up to the  
 2042 time it is completed, canceled, or aborted. Since a Job might already be printing by the time a Cancel-Job is  
 2043 received, some media sheet pages might be printed before the job is actually terminated.

2044 The IPP object **MUST** accept or reject the request based on the job's current state and transition the job to  
 2045 the indicated new state as follows:

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

2046 Rule 1: If the implementation requires some measurable time to cancel the job in the 'processing' or  
 2047 'processing-stopped' job states, the IPP object **MUST** add the 'processing-to-stop-point' value to the job's  
 2048 "job-state-reasons" attribute and then transition the job to the 'canceled' state when the processing ceases  
 2049 (see section 4.3.8).

2050 Rule 2: If the Job object already has the 'processing-to-stop-point' value in its "job-state-reasons" attribute,  
 2051 then the Printer object **MUST** reject a Cancel-Job operation.

2052 *Access Rights:* The authenticated user (see section 8.3) performing this operation must either be the job  
 2053 owner or an operator or administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP  
 2054 object **MUST** reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or  
 2055 'client-error-not-authorized' as appropriate.

#### 2056 3.3.3.1 Cancel-Job Request

2057 The following groups of attributes are part of the Cancel-Job Request:

2058 Group 1: Operation Attributes

2059 Natural Language and Character Set:

2060 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.1.

2061 Target:

2062 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation  
2063 attribute(s) which define the target for this operation as described in section 3.1.5.

2064 Requesting User Name:

2065 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
2066 described in section 8.3.

2067 "message" (text(127)):

2068 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this  
2069 attribute. It is a message to the operator. This "message" attribute is not the same as the "job-  
2070 message-from-operator" attribute. That attribute is used to report a message from the operator to the  
2071 end user that queries that attribute. This "message" operation attribute is used to send a message  
2072 from the client to the operator along with the operation request. It is an implementation decision of  
2073 how or where to display this message to the operator (if at all).

2074 3.3.3.2 Cancel-Job Response

2075 The following sets of attributes are part of the Cancel-Job Response:

2076 Group 1: Operation Attributes

2077 Status Message:

2078 In addition to the REQUIRED status code returned in every response, the response OPTIONALLY  
2079 includes a "status-message" (text(255)) operation attribute as described in sections 13 and 3.1.6.

2080 Natural Language and Character Set:

2081 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.2.

2082 Group 2: Unsupported Attributes

2083 See section 3.1.7 for details on returning Unsupported Attributes.

2084  
2085 Once a successful response has been sent, the implementation guarantees that the Job will eventually end up  
2086 in the 'canceled' state. Between the time of the Cancel-Job operation is accepted and when the job enters the  
2087 'canceled' job-state (see section 4.3.7), the "job-state-reasons" attribute SHOULD contain the 'processing-to-  
2088 stop-point' value which indicates to later queries that although the Job might still be 'processing', it will  
2089 eventually end up in the 'canceled' state, not the 'completed' state.

## 2096 3.3.4 Get-Job-Attributes Operation

2097 This REQUIRED operation allows a client to request the values of attributes of a Job object and it is almost  
2098 identical to the Get-Printer-Attributes operation (see section 3.2.5). The only differences are that the  
2099 operation is directed at a Job object rather than a Printer object, there is no "document-format" operation  
2100 attribute used when querying a Job object, and the returned attribute group is a set of Job object attributes  
2101 rather than a set of Printer object attributes.

2102 For Jobs, the possible names of attribute groups are:

- 2103 - 'job-template': the subset of the Job Template attributes that apply to a Job object (the first column of  
2104 the table in Section 4.2) that the implementation supports for Job objects.
- 2105 - 'job-description': the subset of the Job Description attributes specified in Section 4.3 that the  
2106 implementation supports for Job objects.
- 2107 - 'all': the special group 'all' that includes all attributes that the implementation supports for Job objects.  
2108

2109 Since a client MAY request specific attributes or named groups, there is a potential that there is some  
2110 overlap. For example, if a client requests, 'job-name' and 'job-description', the client is actually requesting  
2111 the "job-name" attribute once by naming it explicitly, and once by inclusion in the 'job-description' group.  
2112 In such cases, the Printer object NEED NOT return the attribute only once in the response even if it is  
2113 requested multiple times. The client SHOULD NOT request the same attribute in multiple ways.

2114 It is NOT REQUIRED that a Job object support all attributes belonging to a group (since some attributes  
2115 are OPTIONAL). However it is REQUIRED that each Job object support all group names.

## 2116 3.3.4.1 Get-Job-Attributes Request

2117 The following groups of attributes are part of the Get-Job-Attributes Request when the request is directed at  
2118 a Job object:

## 2119 Group 1: Operation Attributes

2120 Natural Language and Character Set:

2121 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.1.  
2122

2123 Target:

2124 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation  
2125 attribute(s) which define the target for this operation as described in section 3.1.5.  
2126

2127 Requesting User Name:

2128 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
2129 described in section 8.3.  
2130

2131 "requested-attributes" (1setOf keyword) :

2132 The client OPTIONALLY supplies this attribute. The IPP object MUST support this attribute. It is  
2133 a set of attribute names and/or attribute group names in whose values the requester is interested. If



2134 the client omits this attribute, the IPP object MUST respond as if this attribute had been supplied  
2135 with a value of 'all'.  
2136

### 2137 3.3.4.2 Get-Job-Attributes Response

2138 The Printer object returns the following sets of attributes as part of the Get-Job-Attributes Response:

#### 2139 Group 1: Operation Attributes

##### 2140 Status Message:

2141 In addition to the REQUIRED status code returned in every response, the response OPTIONALLY  
2142 includes a "status-message" (text(255)) operation attribute as described in sections 13 and 3.1.6.

2143

##### 2144 Natural Language and Character Set:

2145 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.2.  
2146 The "attributes-natural-language" MAY be the natural language of the Job object, rather than the  
2147 one requested.

2148

#### 2149 Group 2: Unsupported Attributes

2150 See section 3.1.7 for details on returning Unsupported Attributes.

2151

2152 The response NEED NOT contain the "requested-attributes" operation attribute with any supplied  
2153 values (attribute keywords) that were requested by the client but are not supported by the IPP object.  
2154 If the Printer object does include unsupported attributes referenced in "requested-attributes" and  
2155 such attributes include group names, such as 'all', the unsupported attributes MUST NOT include  
2156 attributes described in the standard but not supported by the implementation.

2157

#### 2158 Group 3: Job Object Attributes

2159 This is the set of requested attributes and their current values. The IPP object ignores (does not  
2160 respond with) any requested attribute or value which is not supported or which is restricted by the  
2161 security policy in force, including whether the requesting user is the user that submitted the job (job  
2162 originating user) or not (see section 8). However, the IPP object MUST respond with the 'unknown'  
2163 value for any supported attribute (including all REQUIRED attributes) for which the IPP object does  
2164 not know the value, unless it would violate the security policy. See the description of the "out-of-  
2165 band" values in the beginning of Section 4.1.

### 2166 3.3.5 Hold-Job Operation

2167 This OPTIONAL operation allows a client to hold a pending job in the queue so that it is not eligible for  
2168 scheduling. If the Hold-Job operation is supported, then the Release-Job operation MUST be supported,  
2169 and vice-versa. The OPTIONAL "job-hold-until" operation attribute allows a client to specify whether to  
2170 hold the job indefinitely or until a specified time period, if supported.

2171 The IPP object **MUST** accept or reject the request based on the job's current state and transition the job to  
 2172 the indicated new state as follows:

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

2173 Rule 1: If the implementation supports multiple reasons for a job to be in the 'pending-held' state, the IPP  
 2174 object **MUST** add the 'job-hold-until-specified' value to the job's "job-state-reasons" attribute.

2175 Rule 2: If the IPP object supports the "job-hold-until" operation attribute, but the specified time period has  
 2176 already started (or is the 'no-hold' value) and there are no other reasons to hold the job, the IPP object  
 2177 **MUST** make the job be a candidate for processing immediately (see Section 4.2.2) by putting the job in the  
 2178 'pending' state.

2179 Note: In order to keep the Hold-Job operation simple, such a request is rejected when the job is in the  
 2180 'processing' or 'processing-stopped' states. If an operation is needed to hold jobs while in these states, it will  
 2181 be added as an additional operation, rather than overloading the Hold-Job operation. Then it is clear to  
 2182 clients by querying the Printer object's "operations-supported" (see Section 4.4.15) and the Job object's  
 2183 "job-state" (see Section 4.3.7) attributes which operations are possible.

2184 *Access Rights:* The authenticated user (see section 8.3) performing this operation must either be the job  
 2185 owner or an operator or administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP  
 2186 object **MUST** reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or  
 2187 'client-error-not-authorized' as appropriate.

### 2188 3.3.5.1 Hold-Job Request

2189 The groups and operation attributes are the same as for a Cancel-Job request (see section 3.3.3.1), with the  
 2190 addition of the following Group 1 Operation attribute:

2191 "job-hold-until" (type3 keyword | name(MAX)):

2192 The client **OPTIONALLY** supplies this Operation attribute. The IPP object **MUST** support this  
 2193 operation attribute in a Hold-Job request, if it supports the "job-hold-until" Job template attribute in  
 2194 create operations. See section 4.2.2. The IPP object **SHOULD** support the "job-hold-until" Job  
 2195 Template attribute for use in job create operations with at least the 'indefinite' value, if it supports  
 2196 the Hold-Job operation. Otherwise, a client cannot create a job and hold it immediately (without  
 2197 picking some supported time period in the future).

2198 If supplied and supported as specified in the Printer's "job-hold-until-supported" attribute, the IPP  
2199 object copies the supplied operation attribute to the Job object, replacing the job's previous "job-  
2200 hold-until" attribute, if present, and makes the job a candidate for scheduling during the supplied  
2201 named time period.

2202 If supplied, but either the "job-hold-until" Operation attribute itself or the value supplied is not  
2203 supported, the IPP object accepts the request, returns the unsupported attribute or value in the  
2204 Unsupported Attributes Group according to section 3.1.7, returns the 'successful-ok-ignored-or-  
2205 substituted-attributes, and holds the job indefinitely until a client performs a subsequent Release-Job  
2206 operation.

2207 If the client (1) supplies a value that specifies a time period that has already started or the 'no-hold'  
2208 value (meaning don't hold the job) and (2) the IPP object supports the "job-hold-until" operation  
2209 attribute and there are no other reasons to hold the job, the IPP object MUST accept the operation  
2210 and make the job be a candidate for processing immediately (see Section 4.2.2).

2211 If the client does not supply a "job-hold-until" Operation attribute in the request, the IPP object  
2212 MUST populate the job object with a "job-hold-until" attribute with the 'indefinite' value (if IPP  
2213 object supports the "job-hold-until" attribute) and hold the job indefinitely, until a client performs a  
2214 Release-Job operation.

### 2215 3.3.5.2 Hold-Job Response

2216 The groups and attributes are the same as for a Cancel-Job response (see section 3.3.3.2).

### 2217 3.3.6 Release-Job Operation

2218 This OPTIONAL operation allows a client to release a previously held job so that it is again eligible for  
2219 scheduling. If the Hold-Job operation is supported, then the Release-Job operation MUST be supported,  
2220 and vice-versa.

2221 This operation removes the "job-hold-until" job attribute, if present, from the job object that had been  
2222 supplied in the create or most recent Hold-Job or Restart-Job operation and remove its effect on the job.  
2223 The IPP object MUST remove the 'job-hold-until-specified' value from the job's "job-state-reasons"  
2224 attribute, if present. See section 4.3.8.

2225 The IPP object MUST accept or reject the request based on the job's current state and transition the job to  
2226 the indicated new state as follows:

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

2227 Rule 1: If there are other reasons to keep the job in the 'pending-held' state, such as 'resources-are-not-  
2228 ready', the job remains in the 'pending-held' state. Thus the 'pending-held' state is not just for jobs that have  
2229 the 'job-hold-until' applied to them, but are for any reason to keep the job from being a candidate for  
2230 scheduling and processing, such as 'resources-are-not-ready'. See the "job-hold-until" attribute (section  
2231 4.2.2).

2232 *Access Rights:* The authenticated user (see section 8.3) performing this operation must either be the job  
2233 owner or an operator or administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP  
2234 object MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or  
2235 'client-error-not-authorized' as appropriate.

2236 The Release-Job Request and Release-Job Response have the same attribute groups and attributes as the  
2237 Cancel-Job operation (see section 3.3.3.1 and 3.3.3.2).

### 2238 3.3.7 Restart-Job Operation

2239 This OPTIONAL operation allows a client to restart a job that is retained in the queue after processing has  
2240 completed (see section 4.3.7.2).

2241 The job is moved to the 'pending' job state and restarts at the beginning on the same IPP Printer object with  
2242 the same attribute values. The Job Description attributes that accumulate job progress, such as "job-  
2243 impressions-completed", "job-media-sheets-completed", and "job-k-octets-processed", MUST be reset to 0  
2244 so that they give an accurate record of the job from its restart point. The job object MUST continue to use  
2245 the same "job-uri" and "job-id" attribute values.

2246 Note: If in the future an operation is needed that does not reset the job progress attributes, then a new  
2247 operation will be defined which makes a copy of the job, assigns a new "job-uri" and "job-id" to the copy  
2248 and resets the job progress attributes in the new copy only.

2249 The IPP object MUST accept or reject the request based on the job's current state, transition the job to the  
2250 indicated new state as follows:

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

2251

2252 Rule 1: If the Job Retention Period has expired for the job in this state, then the IPP object rejects the  
2253 operation. See section 4.3.7.2.

2254 Note: In order to prevent a user from inadvertently restarting a job in the middle, the Restart-Job request is  
2255 rejected when the job is in the 'processing' or 'processing-stopped' states. If in the future an operation is  
2256 needed to hold or restart jobs while in these states, it will be added as an additional operation, rather than  
2257 overloading the Restart-Job operation, so that it is clear that the user intended that the current job not be  
2258 completed.

2259 *Access Rights:* The authenticated user (see section 8.3) performing this operation must either be the job  
2260 owner or an operator or administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP  
2261 object MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or  
2262 'client-error-not-authorized' as appropriate.

### 2263 3.3.7.1 Restart-Job Request

2264 The groups and attributes are the same as for a Cancel-Job request (see section 3.3.3.1), with the addition of  
2265 the following Group 1 Operation attribute:

2266 "job-hold-until" (type3 keyword | name(MAX)):

2267 The client OPTIONALLY supplies this attribute. The IPP object MUST support this Operation  
2268 attribute in a Restart-Job request, if it supports the "job-hold-until" Job Template attribute in create  
2269 operations. See section 4.2.2. Otherwise, the IPP object NEED NOT support the "job-hold-until"  
2270 Operation attribute in a Restart-Job request.

2271 If supplied and supported as specified in the Printer's "job-hold-until-supported" attribute, the IPP  
2272 object copies the supplied Operation attribute to the Job object, replacing the job's previous "job-  
2273 hold-until" attribute, if present, and makes the job a candidate for scheduling during the supplied  
2274 named time period. See section 4.2.2.

2275 If supplied, but the value is not supported, the IPP object accepts the request, returns the  
2276 unsupported attribute or value in the Unsupported Attributes Group according to section 3.1.7,  
2277 returns the 'successful-ok-ignored-or-substituted-attributes' status code, and holds the job  
2278 indefinitely until a client performs a subsequent Release-Job operation.

2279 If supplied, but the "job-hold-until" Operation attribute itself is not supported, the IPP object accepts  
2280 the request, returns the unsupported attribute with the out-of-band 'unsupported' value in the  
2281 Unsupported Attributes Group according to section 3.1.7, returns the 'successful-ok-ignored-or-  
2282 substituted-attributes' status code, and restarts the job, i.e., ignores the "job-hold-until" attribute.

2283 If the client (1) supplies a value that specifies a time period that has already started or the 'no-hold'  
2284 value (meaning don't hold the job) and (2) the IPP object supports the "job-hold-until" operation  
2285 attribute and there are no other reasons to hold the job, the IPP object makes the job a candidate for  
2286 processing immediately (see Section 4.2.2).

2287 If the client does not supply a "job-hold-until" operation attribute in the request, the IPP object  
2288 removes the "job-hold-until" attribute, if present, from the job. If there are no other reasons to hold  
2289 the job, the Restart-Job operation makes the job a candidate for processing immediately (see Section  
2290 4.2.2).

### 2291 3.3.7.2 Restart-Job Response

2292 The groups and attributes are the same as for a Cancel-Job response (see section 3.3.3.2).

2293 Note: In the future an OPTIONAL Modify-Job or Set-Job-Attributes operation may be specified that  
2294 allows the client to modify other attributes before releasing the restarted job.

## 2295 4. Object Attributes

2296 This section describes the attributes with their corresponding attribute syntaxes and values that are part of  
2297 the IPP model. The sections below show the objects and their associated attributes which are included  
2298 within the scope of this protocol. Many of these attributes are derived from other relevant documents:

- 2299 - Document Printing Application (DPA) [ISO10175]
- 2300 - RFC 1759 Printer MIB [RFC1759]

2301

2302 Each attribute is uniquely identified in this document using a "keyword" (see section 12.2.1) which is the  
2303 name of the attribute. The keyword is included in the section header describing that attribute.

2304 Note: Not only are keywords used to identify attributes, but one of the attribute syntaxes described below is  
2305 "keyword" so that some attributes have keyword values. Therefore, these attributes are defined as having  
2306 an attribute syntax that is a set of keywords.

### 2307 4.1 Attribute Syntaxes

2308 This section defines the basic attribute syntax types that all clients and IPP objects MUST be able to accept  
2309 in responses and accept in requests, respectively. Each attribute description in sections 3 and 4 includes the  
2310 name of attribute syntax(es) in the heading (in parentheses). A conforming implementation of an attribute

2311 MUST include the semantics of the attribute syntax(es) so identified. Section 6.3 describes how the  
2312 protocol can be extended with new attribute syntaxes.

2313 The attribute syntaxes are specified in the following sub-sections, where the sub-section heading is the  
2314 keyword name of the attribute syntax inside the single quotes. In operation requests and responses each  
2315 attribute value MUST be represented as one of the attribute syntaxes specified in the sub-section heading  
2316 for the attribute. In addition, the value of an attribute in a response (but not in a request) MAY be one of  
2317 the "out-of-band" values whose special encoding rules are defined in the "Encoding and Transport"  
2318 document [IPP-PRO]. Standard "out-of-band" values are:

2319     `unknown`: The attribute is supported by the IPP object, but the value is unknown to the IPP object for  
2320     some reason.

2321     `unsupported`: The attribute is unsupported by the IPP object. This value MUST be returned only as the  
2322     value of an attribute in the Unsupported Attributes Group.

2323     `no-value`: The attribute is supported by the Printer object, but the administrator has not yet configured a  
2324     value.

2325

2326 All attributes in a request MUST have one or more values as defined in Sections 4.2 to 4.4. Thus clients  
2327 MUST NOT supply attributes with "out-of-band" values. All attributes in a response MUST have one or  
2328 more values as defined in Sections 4.2 to 4.4 or a single "out-of-band" value.

2329 Most attributes are defined to have a single attribute syntax. However, a few attributes (e.g., "job-sheet",  
2330 "media", "job-hold-until") are defined to have several attribute syntaxes, depending on the value. These  
2331 multiple attribute syntaxes are separated by the "|" character in the sub-section heading to indicate the  
2332 choice. Since each value MUST be tagged as to its attribute syntax in the protocol, a single-valued attribute  
2333 instance may have any one of its attribute syntaxes and a multi-valued attribute instance may have a mixture  
2334 of its defined attribute syntaxes.

#### 2335 4.1.1 `text`

2336 A text attribute is an attribute whose value is a sequence of zero or more characters encoded in a maximum  
2337 of 1023 (MAX) octets. MAX is the maximum length for each value of any text attribute. However, if an  
2338 attribute will always contain values whose maximum length is much less than MAX, the definition of that  
2339 attribute will include a qualifier that defines the maximum length for values of that attribute. For example:  
2340 the "printer-location" attribute is specified as "printer-location (text(127))". In this case, text values for  
2341 "printer-location" MUST NOT exceed 127 octets; if supplied with a longer text string via some external  
2342 interface (other than the protocol), implementations are free to truncate to this shorter length limitation.

2343 In this document, all text attributes are defined using the `text` syntax. However, `text` is used only for  
2344 brevity; the formal interpretation of `text` is: `textWithoutLanguage | textWithLanguage`. That is, for any  
2345 attribute defined in this document using the `text` attribute syntax, all IPP objects and clients MUST support  
2346 both the `textWithoutLanguage` and `textWithLanguage` attribute syntaxes. However, in actual usage and  
2347 protocol execution, objects and clients accept and return only one of the two syntax per attribute. The  
2348 syntax `text` never appears "on-the-wire".

2349 Both 'textWithoutLanguage' and 'textWithLanguage' are needed to support the real world needs of  
2350 interoperability between sites and systems that use different natural languages as the basis for human  
2351 communication. Generally, one natural language applies to all text attributes in a given request or response.  
2352 The language is indicated by the "attributes-natural-language" operation attribute defined in section 3.1.4 or  
2353 "attributes-natural-language" job attribute defined in section 4.3.20, and there is no need to identify the  
2354 natural language for each text string on a value-by-value basis. In these cases, the attribute syntax  
2355 'textWithoutLanguage' is used for text attributes. In other cases, the client needs to supply or the Printer  
2356 object needs to return a text value in a natural language that is different from the rest of the text values in  
2357 the request or response. In these cases, the client or Printer object uses the attribute syntax  
2358 'textWithLanguage' for text attributes (this is the Natural Language Override mechanism described in  
2359 section 3.1.4).

2360 The 'textWithoutLanguage' and 'textWithLanguage' attribute syntaxes are described in more detail in the  
2361 following sections.

#### 2362 4.1.1.1 'textWithoutLanguage'

2363 The 'textWithoutLanguage' syntax indicates a value that is sequence of zero or more characters. Text  
2364 strings are encoded using the rules of some charset. The Printer object MUST support the UTF-8 charset  
2365 [RFC2279] and MAY support additional charsets to represent 'text' values, provided that the charsets are  
2366 registered with IANA [IANA-CS]. See Section 4.1.7 for the definition of the 'charset' attribute syntax,  
2367 including restricted semantics and examples of charsets.

#### 2368 4.1.1.2 'textWithLanguage'

2369 The 'textWithLanguage' attribute syntax is a compound attribute syntax consisting of two parts: a  
2370 'textWithoutLanguage' part plus an additional 'naturalLanguage' (see section 4.1.8) part that overrides the  
2371 natural language in force. The 'naturalLanguage' part explicitly identifies the natural language that applies  
2372 to the text part of that value and that value alone. For any give text attribute, the 'textWithoutLanguage' part  
2373 is limited to the maximum length defined for that attribute, but the 'naturalLanguage' part is always limited  
2374 to 63 octets. Using the 'textWithLanguage' attribute syntax rather than the normal 'textWithoutLanguage'  
2375 syntax is the so-called Natural Language Override mechanism and MUST be supported by all IPP objects  
2376 and clients.

2377 If the attribute is multi-valued (1setOf text), then the 'textWithLanguage' attribute syntax MUST be used to  
2378 explicitly specify each attribute value whose natural language needs to be overridden. Other values in a  
2379 multi-valued 'text' attribute in a request or a response revert to the natural language of the operation  
2380 attribute.

2381 In a create request, the Printer object MUST accept and store with the Job object any natural language in the  
2382 "attributes-natural-language" operation attribute, whether the Printer object supports that natural language  
2383 or not. Furthermore, the Printer object MUST accept and store any 'textWithLanguage' attribute value,  
2384 whether the Printer object supports that natural language or not. These requirements are independent of the  
2385 value of the "ipp-attribute-fidelity" operation attribute that the client MAY supply.



2386 Example: If the client supplies the "attributes-natural-language" operation attribute with the value: 'en'  
2387 indicating English, but the value of the "job-name" attribute is in French, the client MUST use the  
2388 'textWithLanguage' attribute syntax with the following two values:

2389 'fr': Natural Language Override indicating French  
2390 'Rapport Mensuel': the job name in French  
2391

2392 See the "Encoding and Transport" document [IPP-PRO] for a detailed example of the 'textWithLanguage'  
2393 attribute syntax.

#### 2394 4.1.2 'name'

2395 This syntax type is used for user-friendly strings, such as a Printer name, that, for humans, are more  
2396 meaningful than identifiers. Names are never translated from one natural language to another. The 'name'  
2397 attribute syntax is essentially the same as 'text', including the REQUIRED support of UTF-8 except that the  
2398 sequence of characters is limited so that its encoded form MUST NOT exceed 255 (MAX) octets.

2399 Also like 'text', 'name' is really an abbreviated notation for either 'nameWithoutLanguage' or  
2400 'nameWithLanguage'. That is, all IPP objects and clients MUST support both the 'nameWithoutLanguage'  
2401 and 'nameWithLanguage' attribute syntaxes. However, in actual usage and protocol execution, objects and  
2402 clients accept and return only one of the two syntax per attribute. The syntax 'name' never appears "on-the-  
2403 wire".

2404 Only the 'text' and 'name' attribute syntaxes permit the Natural Language Override mechanism.

2405 Some attributes are defined as 'type3 keyword | name'. These attributes support values that are either type3  
2406 keywords or names. This dual-syntax mechanism enables a site administrator to extend these attributes to  
2407 legally include values that are locally defined by the site administrator. Such names are not registered with  
2408 IANA.

##### 2409 4.1.2.1 'nameWithoutLanguage'

2410 The 'nameWithoutLanguage' syntax indicates a value that is sequence of zero or more characters so that its  
2411 encoded form does not exceed MAX octets.

##### 2412 4.1.2.2 'nameWithLanguage'

2413 The 'nameWithLanguage' attribute syntax is a compound attribute syntax consisting of two parts: a  
2414 'nameWithoutLanguage' part plus an additional 'naturalLanguage' (see section 4.1.8) part that overrides the  
2415 natural language in force. The 'naturalLanguage' part explicitly identifies the natural language that applies  
2416 to that name value and that name value alone.

2417 The 'nameWithLanguage' attribute syntax behaves the same as the 'textWithLanguage' syntax. If a name is  
2418 in a language that is different than the rest of the object or operation, then this 'nameWithLanguage' syntax  
2419 is used rather than the generic 'nameWithoutLanguage' syntax.

2420 Example: If the client supplies the "attributes-natural-language" operation attribute with the value: 'en'  
2421 indicating English, but the "printer-name" attribute is in German, the client MUST use the  
2422 'nameWithLanguage' attribute syntax as follows:

2423 'de': Natural Language Override indicating German  
2424 'Farbdrucker': the Printer name in German  
2425

#### 2426 4.1.2.3 Matching 'name' attribute values

2427 For purposes of matching two 'name' attribute values for equality, such as in job validation (where a client-  
2428 supplied value for attribute "xxx" is checked to see if the value is among the values of the Printer object's  
2429 corresponding "xxx-supported" attribute), the following match rules apply:

2430 1. 'keyword' values never match 'name' values.

2431 2. 'name' (nameWithoutLanguage and nameWithLanguage) values match if (1) the name parts  
2432 match and (2) the Associated Natural-Language parts (see section 3.1.4.1) match. The matching  
2433 rules are:

2434 a. the name parts match if the two names are identical character by character, except it is  
2435 RECOMMENDED that case be ignored. For example: 'Ajax-letter-head-white' MUST  
2436 match 'Ajax-letter-head-white' and SHOULD match 'ajax-letter-head-white' and 'AJAX-  
2437 LETTER-HEAD-WHITE'.

2438 b. the Associated Natural-Language parts match if the shorter of the two meets the syntactic  
2439 requirements of RFC 1766 [RFC1766] and matches byte for byte with the longer. For  
2440 example, 'en' matches 'en', 'en-us' and 'en-gb', but matches neither 'fr' nor 'e'.

#### 2441 4.1.3 'keyword'

2442 The 'keyword' attribute syntax is a sequence of characters, length: 1 to 255, containing only the US-ASCII  
2443 [ASCII] encoded values for lowercase letters ("a" - "z"), digits ("0" - "9"), hyphen ("-"), dot ("."), and  
2444 underscore ("\_"). The first character MUST be a lowercase letter. Furthermore, keywords MUST be in  
2445 U.S. English.

2446 This syntax type is used for enumerating semantic identifiers of entities in the abstract protocol, i.e., entities  
2447 identified in this document. Keywords are used as attribute names or values of attributes. Unlike 'text' and  
2448 'name' attribute values, 'keyword' values MUST NOT use the Natural Language Override mechanism, since  
2449 they MUST always be US-ASCII and U.S. English.

2450 Keywords are for use in the protocol. A user interface will likely provide a mapping between protocol  
2451 keywords and displayable user-friendly words and phrases which are localized to the natural language of  
2452 the user. While the keywords specified in this document MAY be displayed to users whose natural  
2453 language is U.S. English, they MAY be mapped to other U.S. English words for U.S. English users, since  
2454 the user interface is outside the scope of this document.

2455 In the definition for each attribute of this syntax type, the full set of defined keyword values for that  
2456 attribute are listed.

2457 When a keyword is used to represent an attribute (its name), it MUST be unique within the full scope of all  
2458 IPP objects and attributes. When a keyword is used to represent a value of an attribute, it MUST be unique  
2459 just within the scope of that attribute. That is, the same keyword MUST NOT be used for two different  
2460 values within the same attribute to mean two different semantic ideas. However, the same keyword MAY  
2461 be used across two or more attributes, representing different semantic ideas for each attribute. Section 6.1  
2462 describes how the protocol can be extended with new keyword values. Examples of attribute name  
2463 keywords:

2464 "job-name"  
2465 "attributes-charset"  
2466

2467 Note: This document uses "type1", "type2", and "type3" prefixes to the "keyword" basic syntax to indicate  
2468 different levels of review for extensions (see section 6.1).

#### 2469 4.1.4 'enum'

2470 The 'enum' attribute syntax is an enumerated integer value that is in the range from 1 to 2\*\*31 - 1 (MAX).  
2471 Each value has an associated 'keyword' name. In the definition for each attribute of this syntax type, the full  
2472 set of possible values for that attribute are listed. This syntax type is used for attributes for which there are  
2473 enum values assigned by other standards, such as SNMP MIBs. A number of attribute enum values in this  
2474 document are also used for corresponding attributes in other standards [RFC1759]. This syntax type is not  
2475 used for attributes to which the administrator may assign values. Section 6.1 describes how the protocol  
2476 can be extended with new enum values.

2477 Enum values are for use in the protocol. A user interface will provide a mapping between protocol enum  
2478 values and displayable user-friendly words and phrases which are localized to the natural language of the  
2479 user. While the enum symbols specified in this document MAY be displayed to users whose natural  
2480 language is U.S. English, they MAY be mapped to other U.S. English words for U.S. English users, since  
2481 the user interface is outside the scope of this document.

2482 Note: SNMP MIBs use '2' for 'unknown' which corresponds to the IPP "out-of-band" value 'unknown'. See  
2483 the description of the "out-of-band" values at the beginning of Section 4.1. Therefore, attributes of type  
2484 'enum' start at '3'.

2485 Note: This document uses "type1", "type2", and "type3" prefixes to the "enum" basic syntax to indicate  
2486 different levels of review for extensions (see section 6.1).

#### 2487 4.1.5 'uri'

2488 The 'uri' attribute syntax is any valid Uniform Resource Identifier or URI [RFC2396]. Most often, URIs are  
2489 simply Uniform Resource Locators or URLs. The maximum length of URIs used as values of IPP  
2490 attributes is 1023 octets. Although most other IPP attribute syntax types allow for only lower-cased values,

2491 this attribute syntax type conforms to the case-sensitive and case-insensitive rules specified in [RFC2396].  
2492 See also [IPP-IIG] for a discussion of case in URIs.

#### 2493 4.1.6 'uriScheme'

2494 The 'uriScheme' attribute syntax is a sequence of characters representing a URI scheme according to RFC  
2495 2396 [RFC2396]. Though RFC 2396 requires that the values be case-insensitive, IPP requires all lower  
2496 case values in IPP attributes to simplify comparing by IPP clients and Printer objects.

2497 Standard values for this syntax type are the following keywords:

2498 'ipp': for IPP schemed URIs (e.g., "ipp:...")  
2499 'http': for HTTP schemed URIs (e.g., "http:...")  
2500 'https': for use with HTTPS schemed URIs (e.g., "https:...") (not on IETF standards track)  
2501 'ftp': for FTP schemed URIs (e.g., "ftp:...")  
2502 'mailto': for SMTP schemed URIs (e.g., "mailto:...")  
2503 'file': for file schemed URIs (e.g., "file:...")  
2504

2505 A Printer object MAY support any URI 'scheme' that has been registered with IANA [IANA-MT]. The  
2506 maximum length of URI 'scheme' values used to represent IPP attribute values is 63 octets.

#### 2507 4.1.7 'charset'

2508 The 'charset' attribute syntax is a standard identifier for a charset. A charset is a coded character set and  
2509 encoding scheme. Charsets are used for labeling certain document contents and 'text' and 'name' attribute  
2510 values. The syntax and semantics of this attribute syntax are specified in RFC 2046 [RFC2046] and  
2511 contained in the IANA character-set Registry [IANA-CS] according to the IANA procedures [RFC2278].  
2512 Though RFC 2046 requires that the values be case-insensitive US-ASCII, IPP requires all lower case values  
2513 in IPP attributes to simplify comparing by IPP clients and Printer objects. When a character-set in the  
2514 IANA registry has more than one name (alias), the name labeled as "(preferred MIME name)", if present,  
2515 MUST be used.

2516 The maximum length of 'charset' values used to represent IPP attribute values is 63 octets.

2517 Some examples are:

2518 'utf-8': ISO 10646 Universal Multiple-Octet Coded Character Set (UCS) represented as the UTF-8  
2519 [RFC2279] transfer encoding scheme in which US-ASCII is a subset charset.  
2520 'us-ascii': 7-bit American Standard Code for Information Interchange (ASCII), ANSI X3.4-1986  
2521 [ASCII]. That standard defines US-ASCII, but RFC 2045 [RFC2045] eliminates most of the control  
2522 characters from conformant usage in MIME and IPP.  
2523 'iso-8859-1': 8-bit One-Byte Coded Character Set, Latin Alphabet Nr 1 [ISO8859-1]. That standard  
2524 defines a coded character set that is used by Latin languages in the Western Hemisphere and  
2525 Western Europe. US-ASCII is a subset charset.  
2526 'iso-10646-ucs-2': ISO 10646 Universal Multiple-Octet Coded Character Set (UCS) represented as two  
2527 octets (UCS-2), with the high order octet of each pair coming first (so-called Big Endian integer).

2528

2529 Some attribute descriptions MAY place additional requirements on charset values that may be used, such as  
2530 REQUIRED values that MUST be supported or additional restrictions, such as requiring that the charset  
2531 have US-ASCII as a subset charset.

#### 2532 4.1.8 'naturalLanguage'

2533 The 'naturalLanguage' attribute syntax is a standard identifier for a natural language and optionally a  
2534 country. The values for this syntax type are defined by RFC 1766 [RFC1766]. Though RFC 1766 requires  
2535 that the values be case-insensitive US-ASCII, IPP requires all lower case to simplify comparing by IPP  
2536 clients and Printer objects. Examples include:

2537 'en': for English  
2538 'en-us': for US English  
2539 'fr': for French  
2540 'de': for German

2541

2542 The maximum length of 'naturalLanguage' values used to represent IPP attribute values is 63 octets.

#### 2543 4.1.9 'mimeType'

2544 The 'mimeType' attribute syntax is the Internet Media Type (sometimes called MIME type) as  
2545 defined by RFC 2046 [RFC2046] and registered according to the procedures of RFC 2048 [RFC2048] for  
2546 identifying a document format. The value MAY include a charset parameter, depending on the  
2547 specification of the Media Type in the IANA Registry [IANA-MT]. Although most other IPP syntax types  
2548 allow for only lower-cased values, this syntax type allows for mixed-case values which are case-insensitive.

2549 Examples are:

2550 'text/html': An HTML document  
2551 'text/plain': A plain text document in US-ASCII (RFC 2046 indicates that in the absence of the charset  
2552 parameter MUST mean US-ASCII rather than simply unspecified) [RFC2046].  
2553 'text/plain; charset=US-ASCII': A plain text document in US-ASCII [52, 56].  
2554 'text/plain; charset=ISO-8859-1': A plain text document in ISO 8859-1 (Latin 1) [ISO8859-1].  
2555 'text/plain; charset=utf-8': A plain text document in ISO 10646 represented as UTF-8 [RFC2279]  
2556 'application/postscript': A PostScript document [RFC2046]  
2557 'application/vnd.hp-PCL': A PCL document [IANA-MT] (charset escape sequence embedded in the  
2558 document data)  
2559 'application/pdf': Portable Document Format - see IANA MIME Media Type registry  
2560 'application/octet-stream': Auto-sense - see section 4.1.9.1

2561

##### 2562 4.1.9.1 Application/octet-stream -- Auto-Sensing the document format

2563 One special type is 'application/octet-stream'. If the Printer object supports this value, the Printer object  
2564 MUST be capable of auto-sensing the format of the document data, either as part of the create operation

2565 and/or at document processing time. During auto-sensing, a Printer may determine that the document-data  
2566 has a format that the Printer doesn't recognize. If the Printer determines this problem before returning an  
2567 operation response, it rejects the request and returns the 'client-error-document-format-not-supported' status  
2568 code. If the Printer determines this problem after accepting the request and returning an operation response  
2569 with one of the successful status codes, the Printer adds the 'unsupported-document-format' value to the  
2570 job's "job-state-reasons" attribute.

2571 If the Printer object's default value attribute "document-format-default" is set to 'application/octet-stream',  
2572 the Printer object not only supports auto-sensing of the document format, but will depend on the result of  
2573 applying its auto-sensing when the client does not supply the "document-format" attribute. If the client  
2574 supplies a document format value, the Printer MUST rely on the supplied attribute, rather than trust its  
2575 auto-sensing algorithm. To summarize:

- 2576 1. If the client does not supply a document format value, the Printer MUST rely on its default value  
2577 setting (which may be 'application/octet-stream' indicating an auto-sensing mechanism).
- 2578 2. If the client supplies a value other than 'application/octet-stream', the client is supplying valid  
2579 information about the format of the document data and the Printer object MUST trust the client  
2580 supplied value more than the outcome of applying an automatic format detection mechanism. For  
2581 example, the client may be requesting the printing of a PostScript file as a 'text/plain' document.  
2582 The Printer object MUST print a text representation of the PostScript commands rather than  
2583 interpret the stream of PostScript commands and print the result.
- 2584 3. If the client supplies a value of 'application/octet-stream', the client is indicating that the Printer  
2585 object MUST use its auto-sensing mechanism on the client supplied document data whether auto-  
2586 sensing is the Printer object's default or not.

2588 Note: Since the auto-sensing algorithm is probabilistic, if the client requests both auto-sensing ("document-  
2589 format" set to 'application/octet-stream') and true fidelity ("ipp-attribute-fidelity" set to 'true'), the Printer  
2590 object might not be able to guarantee exactly what the end user intended (the auto-sensing algorithm might  
2591 mistake one document format for another), but it is able to guarantee that its auto-sensing mechanism be  
2592 used.

2593 The maximum length of a 'mimeType' value to represent IPP attribute values is 255 octets.

#### 2594 4.1.10 'octetString'

2595 The 'octetString' attribute syntax is a sequence of octets encoded in a maximum of 1023 octets which is  
2596 indicated in sub-section headers using the notation: octetString(MAX). This syntax type is used for opaque  
2597 data.

#### 2598 4.1.11 'boolean'

2599 The 'boolean' attribute syntax has only two values: 'true' and 'false'.

## 2600 4.1.12 'integer'

2601 The 'integer' attribute syntax is an integer value that is in the range from  $-2^{31}$  (MIN) to  $2^{31} - 1$  (MAX).  
2602 Each individual attribute may specify the range constraint explicitly in sub-section headers if the range is  
2603 different from the full range of possible integer values. For example: job-priority (integer(1:100)) for the  
2604 "job-priority" attribute. However, the enforcement of that additional constraint is up to the IPP objects, not  
2605 the protocol.

## 2606 4.1.13 'rangeOfInteger'

2607 The 'rangeOfInteger' attribute syntax is an ordered pair of integers that defines an inclusive range of integer  
2608 values. The first integer specifies the lower bound and the second specifies the upper bound. If a range  
2609 constraint is specified in the header description for an attribute in this document whose attribute syntax is  
2610 'rangeOfInteger' (i.e., 'X:Y' indicating X as a minimum value and Y as a maximum value), then the  
2611 constraint applies to both integers.

## 2612 4.1.14 'dateTime'

2613 The 'dateTime' attribute syntax is a standard, fixed length, 11 octet representation of the "DateAndTime"  
2614 syntax as defined in RFC 2579 [RFC2579]. RFC 2579 also identifies an 8 octet representation of a  
2615 "DateAndTime" value, but IPP objects MUST use the 11 octet representation. A user interface will provide  
2616 a mapping between protocol dateTime values and displayable user-friendly words or presentation values  
2617 and phrases which are localized to the natural language and date format of the user, including time zone.

## 2618 4.1.15 'resolution'

2619 The 'resolution' attribute syntax specifies a two-dimensional resolution in the indicated units. It consists of  
2620 3 values: a cross feed direction resolution (positive integer value), a feed direction resolution (positive  
2621 integer value), and a units value. The semantics of these three components are taken from the Printer MIB  
2622 [RFC1759] suggested values. That is, the cross feed direction component resolution component is the same  
2623 as the prtMarkerAddressabilityXFeedDir object in the Printer MIB, the feed direction component resolution  
2624 component is the same as the prtMarkerAddressabilityFeedDir in the Printer MIB, and the units component  
2625 is the same as the prtMarkerAddressabilityUnit object in the Printer MIB (namely, '3' indicates dots per inch  
2626 and '4' indicates dots per centimeter). All three values MUST be present even if the first two values are the  
2627 same. Example: '300', '600', '3' indicates a 300 dpi cross-feed direction resolution, a 600 dpi feed direction  
2628 resolution, since a '3' indicates dots per inch (dpi).

## 2629 4.1.16 '1setOf X'

2630 The '1setOf X' attribute syntax is 1 or more values of attribute syntax type X. This syntax type is used for  
2631 multi-valued attributes. The syntax type is called '1setOf' rather than just 'setOf' as a reminder that the set  
2632 of values MUST NOT be empty (i.e., a set of size 0). Sets are normally unordered. However each attribute  
2633 description of this type may specify that the values MUST be in a certain order for that attribute.

## 2634 4.2 Job Template Attributes

2635 Job Template attributes describe job processing behavior. Support for Job Template attributes by a Printer  
2636 object is OPTIONAL (see section 12.2.3 for a description of support for OPTIONAL attributes). Also,  
2637 clients OPTIONALLY supply Job Template attributes in create requests.

2638 Job Template attributes conform to the following rules. For each Job Template attribute called "xxx":

2639 1. If the Printer object supports "xxx" then it MUST support both a "xxx-default" attribute (unless there  
2640 is a "No" in the table below) and a "xxx-supported" attribute. If the Printer object doesn't support  
2641 "xxx", then it MUST support neither an "xxx-default" attribute nor an "xxx-supported" attribute,  
2642 and it MUST treat an attribute "xxx" supplied by a client as unsupported. An attribute "xxx" may be  
2643 supported for some document formats and not supported for other document formats. For example,  
2644 it is expected that a Printer object would only support "orientation-requested" for some document  
2645 formats (such as 'text/plain' or 'text/html') but not others (such as 'application/postscript').  
2646

2647 2. "xxx" is OPTIONALLY supplied by the client in a create request. If "xxx" is supplied, the client is  
2648 indicating a desired job processing behavior for this Job. When "xxx" is not supplied, the client is  
2649 indicating that the Printer object apply its default job processing behavior at job processing time if  
2650 the document content does not contain an embedded instruction indicating an xxx-related behavior.  
2651

2652 Since an administrator MAY change the default value attribute after a Job object has been submitted  
2653 but before it has been processed, the default value used by the Printer object at job processing time  
2654 may be different than the default value in effect at job submission time.  
2655

2656 3. The "xxx-supported" attribute is a Printer object attribute that describes which job processing  
2657 behaviors are supported by that Printer object. A client can query the Printer object to find out what  
2658 xxx-related behaviors are supported by inspecting the returned values of the "xxx-supported"  
2659 attribute.  
2660

2661 Note: The "xxx" in each "xxx-supported" attribute name is singular, even though an "xxx-  
2662 supported" attribute usually has more than one value, such as "job-sheet-supported", unless the  
2663 "xxx" Job Template attribute is plural, such as "finishings" or "sides". In such cases the "xxx-  
2664 supported" attribute names are: "finishings-supported" and "sides-supported".  
2665

2666 4. The "xxx-default" default value attribute describes what will be done at job processing time when no  
2667 other job processing information is supplied by the client (either explicitly as an IPP attribute in the  
2668 create request or implicitly as an embedded instruction within the document data).  
2669

2670 If an application wishes to present an end user with a list of supported values from which to choose, the  
2671 application SHOULD query the Printer object for its supported value attributes. The application SHOULD  
2672 also query the default value attributes. If the application then limits selectable values to only those value  
2673 that are supported, the application can guarantee that the values supplied by the client in the create request  
2674 all fall within the set of supported values at the Printer. When querying the Printer, the client MAY  
2675 enumerate each attribute by name in the Get-Printer-Attributes Request, or the client MAY just name the



2676 "job-template" group in order to get the complete set of supported attributes (both supported and default  
2677 attributes).

2678 The "finishings" attribute is an example of a Job Template attribute. It can take on a set of values such as  
2679 'staple', 'punch', and/or 'cover'. A client can query the Printer object for the "finishings-supported" attribute  
2680 and the "finishings-default" attribute. The supported attribute contains a set of supported values. The  
2681 default value attribute contains the finishing value(s) that will be used for a new Job if the client does not  
2682 supply a "finishings" attribute in the create request and the document data does not contain any  
2683 corresponding finishing instructions. If the client does supply the "finishings" attribute in the create  
2684 request, the IPP object validates the value or values to make sure that they are a subset of the supported  
2685 values identified in the Printer object's "finishings-supported" attribute. See section 3.1.7.

2686 The table below summarizes the names and relationships for all Job Template attributes. The first column  
2687 of the table (labeled "Job Attribute") shows the name and syntax for each Job Template attribute in the Job  
2688 object. These are the attributes that can optionally be supplied by the client in a create request. The last  
2689 two columns (labeled "Printer: Default Value Attribute" and "Printer: Supported Values Attribute") shows  
2690 the name and syntax for each Job Template attribute in the Printer object (the default value attribute and the  
2691 supported values attribute). A "No" in the table means the Printer MUST NOT support the attribute (that is,  
2692 the attribute is simply not applicable). For brevity in the table, the 'text' and 'name' entries do not show the  
2693 maximum length for each attribute.

2694	+=====+	+=====+	+=====+
2695	Job Attribute	Printer: Default Value	Printer: Supported
2696		Attribute	Values Attribute
2697	+=====+	+=====+	+=====+
2698	job-priority	job-priority-default	job-priority-supported
2699	(integer 1:100)	(integer 1:100)	(integer 1:100)
2700	+-----+	+-----+	+-----+
2701	job-hold-until	job-hold-until-	job-hold-until-
2702	(type3 keyword	default	supported
2703	name)	(type3 keyword	(1setOf (
2704		name)	type3 keyword   name))
2705	+-----+	+-----+	+-----+
2706	job-sheets	job-sheets-default	job-sheets-supported
2707	(type3 keyword	(type3 keyword	(1setOf (
2708	name)	name)	type3 keyword   name))
2709	+-----+	+-----+	+-----+
2710	multiple-document-	multiple-document-	multiple-document-
2711	handling	handling-default	handling-supported
2712	(type2 keyword)	(type2 keyword)	(1setOf type2 keyword)
2713	+-----+	+-----+	+-----+
2714	copies	copies-default	copies-supported
2715	(integer (1:MAX))	(integer (1:MAX))	(rangeOfInteger
2716			(1:MAX))
2717	+-----+	+-----+	+-----+
2718	finishings	finishings-default	finishings-supported
2719	(1setOf type2 enum)	(1setOf type2 enum)	(1setOf type2 enum)
2720	+-----+	+-----+	+-----+
2721	page-ranges	No	page-ranges-
2722	(1setOf		supported (boolean)
2723	rangeOfInteger		
2724	(1:MAX))		
2725	+-----+	+-----+	+-----+
2726	sides	sides-default	sides-supported
2727	(type2 keyword)	(type2 keyword)	(1setOf type2 keyword)
2728	+-----+	+-----+	+-----+
2729	number-up	number-up-default	number-up-supported
2730	(integer (1:MAX))	(integer (1:MAX))	(1setOf integer
2731			(1:MAX)
2732			rangeOfInteger
2733			(1:MAX))
2734	+-----+	+-----+	+-----+
2735	orientation-	orientation-requested-	orientation-requested-
2736	requested	default	supported
2737	(type2 enum)	(type2 enum)	(1setOf type2 enum)
2738	+-----+	+-----+	+-----+
2739	media	media-default	media-supported
2740	(type3 keyword	(type3 keyword	(1setOf (
2741	name)	name)	type3 keyword   name))
2742			
2743			media-ready

2744			(1setOf (	
2745			type3 keyword   name))	
2746	+-----+	+-----+	+-----+	+-----+
2747	printer-resolution	printer-resolution-	printer-resolution-	
2748	(resolution)	default	supported	
2749		(resolution)	(1setOf resolution)	
2750	+-----+	+-----+	+-----+	+-----+
2751	print-quality	print-quality-default	print-quality-	
2752	(type2 enum)	(type2 enum)	supported	
2753			(1setOf type2 enum)	
2754	+-----+	+-----+	+-----+	+-----+

2755

2756

#### 2757 4.2.1 job-priority (integer(1:100))

2758 This attribute specifies a priority for scheduling the Job. A higher value specifies a higher priority. The  
 2759 value 1 indicates the lowest possible priority. The value 100 indicates the highest possible priority. Among  
 2760 those jobs that are ready to print, a Printer **MUST** print all jobs with a priority value of n before printing  
 2761 those with a priority value of n-1 for all n.

2762 If the Printer object supports this attribute, it **MUST** always support the full range from 1 to 100. No  
 2763 administrative restrictions are permitted. This way an end-user can always make full use of the entire range  
 2764 with any Printer object. If privileged jobs are implemented outside IPP/1.1, they **MUST** have priorities  
 2765 higher than 100, rather than restricting the range available to end-users.

2766 If the client does not supply this attribute and this attribute is supported by the Printer object, the Printer  
 2767 object **MUST** use the value of the Printer object's "job-priority-default" at job submission time (unlike most  
 2768 Job Template attributes that are used if necessary at job processing time).

2769 The syntax for the "job-priority-supported" is also integer(1:100). This single integer value indicates the  
 2770 number of priority levels supported. The Printer object **MUST** take the value supplied by the client and  
 2771 map it to the closest integer in a sequence of n integers values that are evenly distributed over the range  
 2772 from 1 to 100 using the formula:

2773 
$$\text{roundToNearestInt}((100x+50)/n)$$

2774 where n is the value of "job-priority-supported" and x ranges from 0 through n-1.

2775 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, the  
 2776 sequence of values is: 17, 50 and 83; if n = 10, the sequence of values is: 5, 15, 25, 35, 45, 55, 65, 75, 85,  
 2777 and 95; if n = 100, the sequence of values is: 1, 2, 3, ... 100.

2778 If the value of the Printer object's "job-priority-supported" is 10 and the client supplies values in the range 1  
 2779 to 10, the Printer object maps them to 5, in the range 11 to 20, the Printer object maps them to 15, etc.

## 2780 4.2.2 job-hold-until (type3 keyword | name (MAX))

2781 This attribute specifies the named time period during which the Job **MUST** become a candidate for printing.

2782 Standard keyword values for named time periods are:

2783 'no-hold': immediately, if there are not other reasons to hold the job

2784 'indefinite': - the job is held indefinitely, until a client performs a Release-Job (section 3.3.6)

2785 'day-time': during the day

2786 'evening': evening

2787 'night': night

2788 'weekend': weekend

2789 'second-shift': second-shift (after close of business)

2790 'third-shift': third-shift (after midnight)

2791

2792 An administrator **MUST** associate allowable print times with a named time period (by means outside the  
2793 scope of this IPP/1.1 document). An administrator is encouraged to pick names that suggest the type of  
2794 time period. An administrator **MAY** define additional values using the 'name' or 'keyword' attribute syntax,  
2795 depending on implementation.

2796 If the value of this attribute specifies a time period that is in the future, the Printer **SHOULD** add the 'job-  
2797 hold-until-specified' value to the job's "job-state-reasons" attribute, **MUST** move the job to the 'pending-  
2798 held' state, and **MUST NOT** schedule the job for printing until the specified time-period arrives.

2799 When the specified time period arrives, the Printer **MUST** remove the 'job-hold-until-specified' value from  
2800 the job's "job-state-reason" attribute, if present. If there are no other job state reasons that keep the job in  
2801 the 'pending-held' state, the Printer **MUST** consider the job as a candidate for processing by moving the job  
2802 to the 'pending' state.

2803 If this job attribute value is the named value 'no-hold', or the specified time period has already started, the  
2804 job **MUST** be a candidate for processing immediately.

2805 If the client does not supply this attribute and this attribute is supported by the Printer object, the Printer  
2806 object **MUST** use the value of the Printer object's "job-hold-until-default" at job submission time (unlike  
2807 most Job Template attributes that are used if necessary at job processing time).

## 2808 4.2.3 job-sheets (type3 keyword | name(MAX))

2809 This attribute determines which job start/end sheet(s), if any, **MUST** be printed with a job.

2810 Standard keyword values are:

2811 'none': no job sheet is printed

2812 'standard': one or more site specific standard job sheets are printed, e.g. a single start sheet or both start  
2813 and end sheet is printed

2814

2815 An administrator MAY define additional values using the 'name' or 'keyword' attribute syntax, depending  
2816 on implementation.

2817 The effect of this attribute on jobs with multiple documents MAY be affected by the "multiple-document-  
2818 handling" job attribute (section 4.2.4), depending on the job sheet semantics.

#### 2819 4.2.4 multiple-document-handling (type2 keyword)

2820 This attribute is relevant only if a job consists of two or more documents. This attribute MUST be  
2821 supported if the Printer supports multiple documents per job (see sections 3.2.4 and 3.3.1). The attribute  
2822 controls finishing operations and the placement of one or more print-stream pages into impressions and  
2823 onto media sheets. When the value of the "copies" attribute exceeds 1, it also controls the order in which  
2824 the copies that result from processing the documents are produced. For the purposes of this explanations, if  
2825 "a" represents an instance of document data, then the result of processing the data in document "a" is a  
2826 sequence of media sheets represented by "a(\*)".

2827 Standard keyword values are:

2828 'single-document': If a Job object has multiple documents, say, the document data is called a and b, then  
2829 the result of processing all the document data (a and then b) MUST be treated as a single sequence  
2830 of media sheets for finishing operations; that is, finishing would be performed on the concatenation  
2831 of the sequences a(\*),b(\*). The Printer object MUST NOT force the data in each document instance  
2832 to be formatted onto a new print-stream page, nor to start a new impression on a new media sheet. If  
2833 more than one copy is made, the ordering of the sets of media sheets resulting from processing the  
2834 document data MUST be a(\*), b(\*), a(\*), b(\*), ..., and the Printer object MUST force each copy  
2835 (a(\*),b(\*)) to start on a new media sheet.

2836 'separate-documents-uncollated-copies': If a Job object has multiple documents, say, the document data  
2837 is called a and b, then the result of processing the data in each document instance MUST be treated  
2838 as a single sequence of media sheets for finishing operations; that is, the sets a(\*) and b(\*) would  
2839 each be finished separately. The Printer object MUST force each copy of the result of processing the  
2840 data in a single document to start on a new media sheet. If more than one copy is made, the ordering  
2841 of the sets of media sheets resulting from processing the document data MUST be a(\*), a(\*), ...,  
2842 b(\*), b(\*) ... .

2843 'separate-documents-collated-copies': If a Job object has multiple documents, say, the document data is  
2844 called a and b, then the result of processing the data in each document instance MUST be treated as  
2845 a single sequence of media sheets for finishing operations; that is, the sets a(\*) and b(\*) would each  
2846 be finished separately. The Printer object MUST force each copy of the result of processing the data  
2847 in a single document to start on a new media sheet. If more than one copy is made, the ordering of  
2848 the sets of media sheets resulting from processing the document data MUST be a(\*), b(\*), a(\*), b(\*),  
2849 ... .

2850 'single-document-new-sheet': Same as 'single-document', except that the Printer object MUST ensure  
2851 that the first impression of each document instance in the job is placed on a new media sheet. This  
2852 value allows multiple documents to be stapled together with a single staple where each document  
2853 starts on a new sheet.

2854

2855 The 'single-document' value is the same as 'separate-documents-collated-copies' with respect to ordering of  
 2856 print-stream pages, but not media sheet generation, since 'single-document' will put the first page of the  
 2857 next document on the back side of a sheet if an odd number of pages have been produced so far for the job,  
 2858 while 'separate-documents-collated-copies' always forces the next document or document copy on to a new  
 2859 sheet. In addition, if the "finishings" attribute specifies 'staple', then with 'single-document', documents a  
 2860 and b are stapled together as a single document with no regard to new sheets, with 'single-document-new-  
 2861 sheet', documents a and b are stapled together as a single document, but document b starts on a new sheet,  
 2862 but with 'separate-documents-uncollated-copies' and 'separate-documents-collated-copies', documents a and  
 2863 b are stapled separately.

2864 Note: None of these values provide means to produce uncollated sheets within a document, i.e., where  
 2865 multiple copies of sheet n are produced before sheet n+1 of the same document.

2866 The relationship of this attribute and the other attributes that control document processing is described in  
 2867 section 15.3.

#### 2868 4.2.5 copies (integer(1:MAX))

2869 This attribute specifies the number of copies to be printed.

2870 On many devices the supported number of collated copies will be limited by the number of physical output  
 2871 bins on the device, and may be different from the number of uncollated copies which can be supported.

2872 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-document-  
 2873 handling" job attribute (section 4.2.4) and the relationship of this attribute and the other attributes that  
 2874 control document processing is described in section 15.3.

#### 2875 4.2.6 finishings (1setOf type2 enum)

2876 This attribute identifies the finishing operations that the Printer uses for each copy of each printed  
 2877 document in the Job. For Jobs with multiple documents, the "multiple-document-handling" attribute  
 2878 determines what constitutes a "copy" for purposes of finishing.

2879 Standard enum values are:

2880 Value	Symbolic Name and Description
2881 '3'	'none': Perform no finishing
2883 '4'	'staple': Bind the document(s) with one or more staples. The exact number and placement of 2884 the staples is site-defined.
2885 '5'	'punch': This value indicates that holes are required in the finished document. The exact 2886 number and placement of the holes is site-defined. The punch specification MAY be 2887 satisfied (in a site- and implementation-specific manner) either by drilling/punching, 2888 or by substituting pre-drilled media.

- 2889 '6' 'cover': This value is specified when it is desired to select a non-printed (or pre-printed)  
2890 cover for the document. This does not supplant the specification of a printed cover  
2891 (on cover stock medium) by the document itself.
- 2892 '7' 'bind': This value indicates that a binding is to be applied to the document; the type and  
2893 placement of the binding is site-defined.
- 2894
- 2895 '8' 'saddle-stitch': Bind the document(s) with one or more staples (wire stitches) along the  
2896 middle fold. The exact number and placement of the staples and the middle fold is  
2897 implementation and/or site-defined.
- 2898 '9' 'edge-stitch': Bind the document(s) with one or more staples (wire stitches) along one edge.  
2899 The exact number and placement of the staples is implementation and/or site-  
2900 defined.
- 2901 '10'-'19' reserved for future generic finishing enum values.

2902 The following values are more specific; they indicate a corner or an edge as if the document were a portrait  
2903 document (see below):

- 2904 '20' 'staple-top-left': Bind the document(s) with one or more staples in the top left corner.
- 2905 '21' 'staple-bottom-left': Bind the document(s) with one or more staples in the bottom left  
2906 corner.
- 2907 '22' 'staple-top-right': Bind the document(s) with one or more staples in the top right corner.
- 2908 '23' 'staple-bottom-right': Bind the document(s) with one or more staples in the bottom right  
2909 corner.
- 2910 '24' 'edge-stitch-left': Bind the document(s) with one or more staples (wire stitches) along the  
2911 left edge. The exact number and placement of the staples is implementation and/or  
2912 site-defined.
- 2913 '25' 'edge-stitch-top': Bind the document(s) with one or more staples (wire stitches) along the  
2914 top edge. The exact number and placement of the staples is implementation and/or  
2915 site-defined.
- 2916 '26' 'edge-stitch-right': Bind the document(s) with one or more staples (wire stitches) along the  
2917 right edge. The exact number and placement of the staples is implementation and/or  
2918 site-defined.
- 2919 '27' 'edge-stitch-bottom': Bind the document(s) with one or more staples (wire stitches) along  
2920 the bottom edge. The exact number and placement of the staples is implementation  
2921 and/or site-defined.
- 2922 '28' 'staple-dual-left': Bind the document(s) with two staples (wire stitches) along the left edge  
2923 assuming a portrait document (see above).
- 2924 '29' 'staple-dual-top': Bind the document(s) with two staples (wire stitches) along the top edge  
2925 assuming a portrait document (see above).
- 2926 '30' 'staple-dual-right': Bind the document(s) with two staples (wire stitches) along the right  
2927 edge assuming a portrait document (see above).
- 2928 '31' 'staple-dual-bottom': Bind the document(s) with two staples (wire stitches) along the bottom  
2929 edge assuming a portrait document (see above).

2930 The 'staple-xxx' values are specified with respect to the document as if the document were a portrait  
2931 document. If the document is actually a landscape or a reverse-landscape document, the client supplies the  
2932 appropriate transformed value. For example, to position a staple in the upper left hand corner of a

2933 landscape document when held for reading, the client supplies the 'staple-bottom-left' value (since  
2934 landscape is defined as a +90 degree rotation from portrait, i.e., anti-clockwise). On the other hand, to  
2935 position a staple in the upper left hand corner of a reverse-landscape document when held for reading, the  
2936 client supplies the 'staple-top-right' value (since reverse-landscape is defined as a -90 degree rotation from  
2937 portrait, i.e., clockwise).

2938 The angle (vertical, horizontal, angled) of each staple with respect to the document depends on the  
2939 implementation which may in turn depend on the value of the attribute.

2940 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-document-  
2941 handling" job attribute (section 4.2.4) and the relationship of this attribute and the other attributes that  
2942 control document processing is described in section 15.3.

2943 If the client supplies a value of 'none' along with any other combination of values, it is the same as if only  
2944 that other combination of values had been supplied (that is the 'none' value has no effect).

#### 2945 4.2.7 page-ranges (1setOf rangeOfInteger (1:MAX))

2946 This attribute identifies the range(s) of print-stream pages that the Printer object uses for each copy of each  
2947 document which are to be printed. Nothing is printed for any pages identified that do not exist in the  
2948 document(s). Ranges MUST be in ascending order, for example: 1-3, 5-7, 15-19 and MUST NOT overlap,  
2949 so that a non-spooling Printer object can process the job in a single pass. If the ranges are not ascending or  
2950 are overlapping, the IPP object MUST reject the request and return the 'client-error-bad-request' status code.  
2951 The attribute is associated with print-stream pages not application-numbered pages (for example, the page  
2952 numbers found in the headers and or footers for certain word processing applications).

2953 For Jobs with multiple documents, the "multiple-document-handling" attribute determines what constitutes  
2954 a "copy" for purposes of the specified page range(s). When "multiple-document-handling" is 'single-  
2955 document', the Printer object MUST apply each supplied page range once to the concatenation of the print-  
2956 stream pages. For example, if there are 8 documents of 10 pages each, the page-range '41:60' prints the  
2957 pages in the 5th and 6th documents as a single document and none of the pages of the other documents are  
2958 printed. When "multiple-document-handling" is 'separate-documents-uncollated-copies' or 'separate-  
2959 documents-collated-copies', the Printer object MUST apply each supplied page range repeatedly to each  
2960 document copy. For the same job, the page-range '1:3, 10:10' would print the first 3 pages and the 10th  
2961 page of each of the 8 documents in the Job, as 8 separate documents.

2962 In most cases, the exact pages to be printed will be generated by a device driver and this attribute would not  
2963 be required. However, when printing an archived document which has already been formatted, the end user  
2964 may elect to print just a subset of the pages contained in the document. In this case, if page-range = n.m is  
2965 specified, the first page to be printed will be page n. All subsequent pages of the document will be printed  
2966 through and including page m.

2967 "page-ranges-supported" is a boolean value indicating whether or not the printer is capable of supporting  
2968 the printing of page ranges. This capability may differ from one PDL to another. There is no "page-ranges-  
2969 default" attribute. If the "page-ranges" attribute is not supplied by the client, all pages of the document will  
2970 be printed.



2971 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-document-  
 2972 handling" job attribute (section 4.2.4) and the relationship of this attribute and the other attributes that  
 2973 control document processing is described in section 15.3.

#### 2974 4.2.8 sides (type2 keyword)

2975 This attribute specifies how print-stream pages are to be imposed upon the sides of an instance of a selected  
 2976 medium, i.e., an impression.

2977 The standard keyword values are:

2978 'one-sided': imposes each consecutive print-stream page upon the same side of consecutive media  
 2979 sheets.

2980 'two-sided-long-edge': imposes each consecutive pair of print-stream pages upon front and back sides of  
 2981 consecutive media sheets, such that the orientation of each pair of print-stream pages on the medium  
 2982 would be correct for the reader as if for binding on the long edge. This imposition is sometimes  
 2983 called 'duplex' or 'head-to-head'.

2984 'two-sided-short-edge': imposes each consecutive pair of print-stream pages upon front and back sides  
 2985 of consecutive media sheets, such that the orientation of each pair of print-stream pages on the  
 2986 medium would be correct for the reader as if for binding on the short edge. This imposition is  
 2987 sometimes called 'tumble' or 'head-to-toe'.

2988

2989 'two-sided-long-edge', 'two-sided-short-edge', 'tumble', and 'duplex' all work the same for portrait or  
 2990 landscape. However 'head-to-toe' is 'tumble' in portrait but 'duplex' in landscape. 'head-to-head' also  
 2991 switches between 'duplex' and 'tumble' when using portrait and landscape modes.

2992 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-document-  
 2993 handling" job attribute (section 4.2.4) and the relationship of this attribute and the other attributes that  
 2994 control document processing is described in section 15.3.

#### 2995 4.2.9 number-up (integer(1:MAX))

2996 This attribute specifies the number of print-stream pages to impose upon a single side of an instance of a  
 2997 selected medium. For example, if the value is:

2998 Value	Description
3000 '1'	the Printer MUST place one print-stream page on a single side of an instance of the selected 3001 medium (MAY add some sort of translation, scaling, or rotation).
3002 '2'	the Printer MUST place two print-stream pages on a single side of an instance of the selected 3003 medium (MAY add some sort of translation, scaling, or rotation).
3004 '4'	the Printer MUST place four print-stream pages on a single side of an instance of the 3005 selected medium (MAY add some sort of translation, scaling, or rotation).

3006

3007 This attribute primarily controls the translation, scaling and rotation of print-stream pages.

3008 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-document-  
 3009 handling" job attribute (section 4.2.4) and the relationship of this attribute and the other attributes that  
 3010 control document processing is described in section 15.3.

#### 3011 4.2.10 orientation-requested (type2 enum)

3012 This attribute indicates the desired orientation for printed print-stream pages; it does not describe the  
 3013 orientation of the client-supplied print-stream pages.

3014 For some document formats (such as 'application/postscript'), the desired orientation of the print-stream  
 3015 pages is specified within the document data. This information is generated by a device driver prior to the  
 3016 submission of the print job. Other document formats (such as 'text/plain') do not include the notion of  
 3017 desired orientation within the document data. In the latter case it is possible for the Printer object to bind  
 3018 the desired orientation to the document data after it has been submitted. It is expected that a Printer object  
 3019 would only support "orientations-requested" for some document formats (e.g., 'text/plain' or 'text/html') but  
 3020 not others (e.g., 'application/postscript'). This is no different than any other Job Template attribute since  
 3021 section 4.2, item 1, points out that a Printer object may support or not support any Job Template attribute  
 3022 based on the document format supplied by the client. However, a special mention is made here since it is  
 3023 very likely that a Printer object will support "orientation-requested" for only a subset of the supported  
 3024 document formats.

3025 Standard enum values are:

3026	Value	Symbolic Name and Description
3027		
3028	'3'	'portrait': The content will be imaged across the short edge of the medium.
3029	'4'	'landscape': The content will be imaged across the long edge of the medium. Landscape is
3030		defined to be a rotation of the print-stream page to be imaged by +90 degrees with
3031		respect to the medium (i.e. anti-clockwise) from the portrait orientation. Note: The
3032		+90 direction was chosen because simple finishing on the long edge is the same edge
3033		whether portrait or landscape
3034	'5'	'reverse-landscape': The content will be imaged across the long edge of the medium.
3035		Reverse-landscape is defined to be a rotation of the print-stream page to be imaged
3036		by -90 degrees with respect to the medium (i.e. clockwise) from the portrait
3037		orientation. Note: The 'reverse-landscape' value was added because some
3038		applications rotate landscape -90 degrees from portrait, rather than +90 degrees.
3039	'6'	'reverse-portrait': The content will be imaged across the short edge of the medium. Reverse-
3040		portrait is defined to be a rotation of the print-stream page to be imaged by 180
3041		degrees with respect to the medium from the portrait orientation. Note: The 'reverse-
3042		portrait' value was added for use with the "finishings" attribute in cases where the
3043		opposite edge is desired for finishing a portrait document on simple finishing devices
3044		that have only one finishing position. Thus a 'text/plain' portrait document can be
3045		stapled "on the right" by a simple finishing device as is common use with some
3046		middle eastern languages such as Hebrew.
3047		

3048 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-document-  
3049 handling" job attribute (section 4.2.4) and the relationship of this attribute and the other attributes that  
3050 control document processing is described in section 15.3.

#### 3051 4.2.11 media (type3 keyword | name(MAX))

3052 This attribute identifies the medium that the Printer uses for all impressions of the Job.

3053 The values for "media" include medium-names, medium-sizes, input-trays and electronic forms so that one  
3054 attribute specifies the media. If a Printer object supports a medium name as a value of this attribute, such a  
3055 medium name implicitly selects an input-tray that contains the specified medium. If a Printer object  
3056 supports a medium size as a value of this attribute, such a medium size implicitly selects a medium name  
3057 that in turn implicitly selects an input-tray that contains the medium with the specified size. If a Printer  
3058 object supports an input-tray as the value of this attribute, such an input-tray implicitly selects the medium  
3059 that is in that input-tray at the time the job prints. This case includes manual-feed input-trays. If a Printer  
3060 object supports an electronic form as the value of this attribute, such an electronic form implicitly selects a  
3061 medium-name that in turn implicitly selects an input-tray that contains the medium specified by the  
3062 electronic form. The electronic form also implicitly selects an image that the Printer MUST merge with the  
3063 document data as its prints each page.

3064 Standard keyword values are (taken from ISO DPA and the Printer MIB) and are listed in section 14. An  
3065 administrator MAY define additional values using the 'name' or 'keyword' attribute syntax, depending on  
3066 implementation.

3067 There is also an additional Printer attribute named "media-ready" which differs from "media-supported" in  
3068 that legal values only include the subset of "media-supported" values that are physically loaded and ready  
3069 for printing with no operator intervention required. If an IPP object supports "media-supported", it NEED  
3070 NOT support "media-ready".

3071 The relationship of this attribute and the other attributes that control document processing is described in  
3072 section 15.3.

#### 3073 4.2.12 printer-resolution (resolution)

3074 This attribute identifies the resolution that Printer uses for the Job.

#### 3075 4.2.13 print-quality (type2 enum)

3076 This attribute specifies the print quality that the Printer uses for the Job.

3077 The standard enum values are:

3078	Value	Symbolic Name and Description
3079		
3080	'3'	'draft': lowest quality available on the printer
3081	'4'	'normal': normal or intermediate quality on the printer

3082       `5'       `high`: highest quality available on the printer  
3083

#### 3084 4.3 Job Description Attributes

3085 The attributes in this section form the attribute group called "job-description". The following table  
3086 summarizes these attributes. The third column indicates whether the attribute is a REQUIRED attribute  
3087 that MUST be supported by Printer objects. If it is not indicated as REQUIRED, then it is OPTIONAL.  
3088 The maximum size in octets for `text` and `name` attributes is indicated in parentheses.

	Attribute	Syntax	REQUIRED?
3089			
3090			
3091			
3092	job-uri	uri	REQUIRED
3093			
3094	job-id	integer(1:MAX)	REQUIRED
3095			
3096	job-printer-uri	uri	REQUIRED
3097			
3098	job-more-info	uri	
3099			
3100	job-name	name (MAX)	REQUIRED
3101			
3102	job-originating-user-name	name (MAX)	REQUIRED
3103			
3104	job-state	type1 enum	REQUIRED
3105			
3106	job-state-reasons	1setOf type2 keyword	REQUIRED
3107			
3108	job-state-message	text (MAX)	
3109			
3110	number-of-documents	integer (0:MAX)	
3111			
3112	output-device-assigned	name (127)	
3113			
3114	time-at-creation	integer (MIN:MAX)	REQUIRED
3115			
3116	time-at-processing	integer (MIN:MAX)	REQUIRED
3117			
3118	time-at-completed	integer (MIN:MAX)	REQUIRED
3119			
3120	job-printer-up-time	integer (1:MAX)	REQUIRED
3121			
3122	date-time-at-creation	dateTime	OPTIONAL
3123			
3124	date-time-at-processing	dateTime	OPTIONAL
3125			
3126	date-time-at-completed	dateTime	OPTIONAL
3127			
3128	number-of-intervening-jobs	integer (0:MAX)	
3129			
3130	job-message-from-operator	text (127)	
3131			
3132	job-k-octets	integer (0:MAX)	
3133			
3134	job-impressions	integer (0:MAX)	
3135			
3136	job-media-sheets	integer (0:MAX)	
3137			
3138	job-k-octets-processed	integer (0:MAX)	

3139	+-----+-----+-----+
3140	job-impressions-completed   integer (0:MAX)
3141	+-----+-----+-----+
3142	job-media-sheets-completed   integer (0:MAX)
3143	+-----+-----+-----+
3144	attributes-charset   charset   REQUIRED
3145	+-----+-----+-----+
3146	attributes-natural-language   naturalLanguage   REQUIRED
3147	+-----+-----+-----+
3148	
3149	

#### 3150 4.3.1 job-uri (uri)

3151 This REQUIRED attribute contains the URI for the job. The Printer object, on receipt of a new job,  
 3152 generates a URI which identifies the new Job. The Printer object returns the value of the "job-uri" attribute  
 3153 as part of the response to a create request. The precise format of a Job URI is implementation dependent.  
 3154 If the Printer object supports more than one URI and there is some relationship between the newly formed  
 3155 Job URI and the Printer object's URI, the Printer object uses the Printer URI supplied by the client in the  
 3156 create request. For example, if the create request comes in over a secure channel, the new Job URI MUST  
 3157 use the same secure channel. This can be guaranteed because the Printer object is responsible for  
 3158 generating the Job URI and the Printer object is aware of its security configuration and policy as well as the  
 3159 Printer URI used in the create request.

3160 For a description of this attribute and its relationship to "job-id" and "job-printer-uri" attribute, see the  
 3161 discussion in section 2.4 on "Object Identity".

#### 3162 4.3.2 job-id (integer(1:MAX))

3163 This REQUIRED attribute contains the ID of the job. The Printer, on receipt of a new job, generates an ID  
 3164 which identifies the new Job on that Printer. The Printer returns the value of the "job-id" attribute as part of  
 3165 the response to a create request. The 0 value is not included to allow for compatibility with SNMP index  
 3166 values which also cannot be 0.

3167 For a description of this attribute and its relationship to "job-uri" and "job-printer-uri" attribute, see the  
 3168 discussion in section 2.4 on "Object Identity".

#### 3169 4.3.3 job-printer-uri (uri)

3170 This REQUIRED attribute identifies the Printer object that created this Job object. When a Printer object  
 3171 creates a Job object, it populates this attribute with the Printer object URI that was used in the create  
 3172 request. This attribute permits a client to identify the Printer object that created this Job object when only  
 3173 the Job object's URI is available to the client. The client queries the creating Printer object to determine  
 3174 which languages, charsets, operations, are supported for this Job.

3175 For a description of this attribute and its relationship to "job-uri" and "job-id" attribute, see the discussion in  
 3176 section 2.4 on "Object Identity".

## 3177 4.3.4 job-more-info (uri)

3178 Similar to "printer-more-info", this attribute contains the URI referencing some resource with more  
3179 information about this Job object, perhaps an HTML page containing information about the Job.

## 3180 4.3.5 job-name (name(MAX))

3181 This REQUIRED attribute is the name of the job. It is a name that is more user friendly than the "job-uri"  
3182 attribute value. It does not need to be unique between Jobs. The Job's "job-name" attribute is set to the  
3183 value supplied by the client in the "job-name" operation attribute in the create request (see Section 3.2.1.1).  
3184 If, however, the "job-name" operation attribute is not supplied by the client in the create request, the Printer  
3185 object, on creation of the Job, MUST generate a name. The printer SHOULD generate the value of the  
3186 Job's "job-name" attribute from the first of the following sources that produces a value: 1) the "document-  
3187 name" operation attribute of the first (or only) document, 2) the "document-URI" attribute of the first (or  
3188 only) document, or 3) any other piece of Job specific and/or Document Content information.

## 3189 4.3.6 job-originating-user-name (name(MAX))

3190 This REQUIRED attribute contains the name of the end user that submitted the print job. The Printer  
3191 object sets this attribute to the most authenticated printable name that it can obtain from the authentication  
3192 service over which the IPP operation was received. Only if such is not available, does the Printer object use  
3193 the value supplied by the client in the "requesting-user-name" operation attribute of the create operation  
3194 (see Section 8).

3195 Note: The Printer object needs to keep an internal originating user id of some form, typically as a credential  
3196 of a principal, with the Job object. Since such an internal attribute is implementation-dependent and not of  
3197 interest to clients, it is not specified as a Job Description attribute. This originating user id is used for  
3198 authorization checks (if any) on all subsequent operation.

## 3199 4.3.7 job-state (type1 enum)

3200 This REQUIRED attribute identifies the current state of the job. Even though the IPP protocol defines  
3201 seven values for job states (plus the out-of-band 'unknown' value - see Section 4.1), implementations only  
3202 need to support those states which are appropriate for the particular implementation. In other words, a  
3203 Printer supports only those job states implemented by the output device and available to the Printer object  
3204 implementation.

3205 Standard enum values are:

3206 Values Symbolic Name and Description

3207

3208 '3' 'pending': The job is a candidate to start processing, but is not yet processing.

3209

3210 '4' 'pending-held': The job is not a candidate for processing for any number of reasons but will  
3211 return to the 'pending' state as soon as the reasons are no longer present. The job's

3212 "job-state-reason" attribute MUST indicate why the job is no longer a candidate for  
3213 processing.

3214  
3215 '5' 'processing': One or more of:

- 3216  
3217 1. the job is using, or is attempting to use, one or more purely software processes  
3218 that are analyzing, creating, or interpreting a PDL, etc.,  
3219 2. the job is using, or is attempting to use, one or more hardware devices that are  
3220 interpreting a PDL, making marks on a medium, and/or performing finishing, such as  
3221 stapling, etc.,  
3222 3. the Printer object has made the job ready for printing, but the output device is not  
3223 yet printing it, either because the job hasn't reached the output device or because the  
3224 job is queued in the output device or some other spooler, awaiting the output device  
3225 to print it.  
3226

3227 When the job is in the 'processing' state, the entire job state includes the detailed  
3228 status represented in the Printer object's "printer-state", "printer-state-reasons", and  
3229 "printer-state-message" attributes.

3230 Implementations MAY, though they NEED NOT, include additional values in the  
3231 job's "job-state-reasons" attribute to indicate the progress of the job, such as adding  
3232 the 'job-printing' value to indicate when the output device is actually making marks  
3233 on paper and/or the 'processing-to-stop-point' value to indicate that the IPP object is  
3234 in the process of canceling or aborting the job. Most implementations won't bother  
3235 with this nuance.  
3236

3237 '6' 'processing-stopped': The job has stopped while processing for any number of reasons and  
3238 will return to the 'processing' state as soon as the reasons are no longer present.  
3239

3240 The job's "job-state-reason" attribute MAY indicate why the job has stopped  
3241 processing. For example, if the output device is stopped, the 'printer-stopped' value  
3242 MAY be included in the job's "job-state-reasons" attribute.  
3243

3244 Note: When an output device is stopped, the device usually indicates its condition in  
3245 human readable form locally at the device. A client can obtain more complete device  
3246 status remotely by querying the Printer object's "printer-state", "printer-state-reasons"  
3247 and "printer-state-message" attributes.  
3248

3249 '7' 'canceled': The job has been canceled by a Cancel-Job operation and the Printer object has  
3250 completed canceling the job and all job status attributes have reached their final  
3251 values for the job. While the Printer object is canceling the job, the job remains in its  
3252 current state, but the job's "job-state-reasons" attribute SHOULD contain the  
3253 'processing-to-stop-point' value and one of the 'canceled-by-user', 'canceled-by-  
3254 operator', or 'canceled-at-device' value. When the job moves to the 'canceled' state,



3255 the 'processing-to-stop-point' value, if present, MUST be removed, but the 'canceled-  
3256 by-xxx', if present, MUST remain.

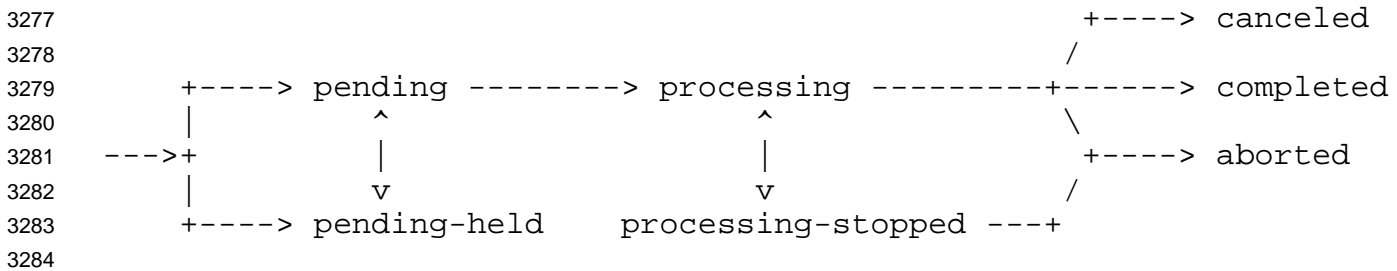
3257  
3258 '8' 'aborted': The job has been aborted by the system, usually while the job was in the  
3259 'processing' or 'processing-stopped' state and the Printer has completed aborting the  
3260 job and all job status attributes have reached their final values for the job. While the  
3261 Printer object is aborting the job, the job remains in its current state, but the job's  
3262 "job-state-reasons" attribute SHOULD contain the 'processing-to-stop-point' and  
3263 'aborted-by-system' values. When the job moves to the 'aborted' state, the  
3264 'processing-to-stop-point' value, if present, MUST be removed, but the 'aborted-by-  
3265 system' value, if present, MUST remain.

3266  
3267 '9' 'completed': The job has completed successfully or with warnings or errors after processing  
3268 and all of the job media sheets have been successfully stacked in the appropriate  
3269 output bin(s) and all job status attributes have reached their final values for the job.  
3270 The job's "job-state-reasons" attribute SHOULD contain one of: 'completed-  
3271 successfully', 'completed-with-warnings', or 'completed-with-errors' values.

3272

3273 The final value for this attribute MUST be one of: 'completed', 'canceled', or 'aborted' before the Printer  
3274 removes the job altogether. The length of time that jobs remain in the 'canceled', 'aborted', and 'completed'  
3275 states depends on implementation. See section 4.3.7.2.

3276 The following figure shows the normal job state transitions.



3285 Normally a job progresses from left to right. Other state transitions are unlikely, but are not forbidden. Not  
3286 shown are the transitions to the 'canceled' state from the 'pending', 'pending-held', and 'processing-stopped'  
3287 states.

3288 Jobs reach one of the three terminal states: 'completed', 'canceled', or 'aborted', after the jobs have  
3289 completed all activity, including stacking output media, after the jobs have completed all activity, and all  
3290 job status attributes have reached their final values for the job.

### 3291 4.3.7.1 Forwarding Servers

3292 As with all other IPP attributes, if the implementation cannot determine the correct value for this attribute,  
3293 it SHOULD respond with the out-of-band value 'unknown' (see section 4.1) rather than try to guess at some  
3294 possibly incorrect value and give the end user the wrong impression about the state of the Job object. For  
3295 example, if the implementation is just a gateway into some printing system from which it can normally get

3296 status, but temporarily is unable, then the implementation should return the 'unknown' value. However, if  
3297 the implementation is a gateway to a printing system that never provides detailed status about the print job,  
3298 the implementation MAY set the IPP Job object's state to 'completed', provided that it also sets the 'queued-  
3299 in-device' value in the job's "job-state-reasons" attribute (see section 4.3.8).

#### 3300 4.3.7.2 Partitioning of Job States

3301 This section partitions the 7 job states into phases: Job Not Completed, Job Retention, Job History, and Job  
3302 Removal. This section also explains the 'job-restartable' value of the "job-state-reasons" Job Description  
3303 attribute for use with the Restart-Job operation.

3304 Job Not Completed: When a job is in the 'pending', 'pending-held', 'processing', or 'processing-stopped'  
3305 states, the job is not completed.

3306 Job Retention: When a job enters one of the three terminal job states: 'completed', 'canceled', or 'aborted',  
3307 the IPP Printer object MAY "retain" the job in a restartable condition for an implementation-defined time  
3308 period. This time period MAY be zero seconds and MAY depend on the terminal job state. This phase is  
3309 called Job Retention. While in the Job Retention phase, the job's document data is retained and a client  
3310 may restart the job using the Restart-Job operation. If the IPP object supports the Restart-Job operation,  
3311 then it SHOULD indicate that the job is restartable by adding the 'job-restartable' value to the job's "job-  
3312 state-reasons" attribute (see Section 4.3.8) during the Job Retention phase.

3313 Job History: After the Job Retention phase expires for a job, the Printer object deletes the document data  
3314 for the job and the job becomes part of the Job History. The Printer object MAY also delete any number of  
3315 the job attributes. Since the job is no longer restartable, the Printer object MUST remove the 'job-  
3316 restartable' value from the job's "job-state-reasons" attribute, if present.

3317 Job Removal: After the job has remained in the Job History for an implementation-defined time, such as  
3318 when the number of jobs exceeds a fixed number or after a fixed time period (which MAY be zero  
3319 seconds), the IPP Printer removes the job from the system.

3320 Using the Get-Jobs operation and supplying the 'not-completed' value for the "which-jobs" operation  
3321 attribute, a client is requesting jobs in the Job Not Completed phase. Using the Get-Jobs operation and  
3322 supplying the 'completed' value for the "which-jobs" operation attribute, a client is requesting jobs in the  
3323 Job Retention and Job History phases. Using the Get-Job-Attributes operation, a client is requesting a job  
3324 in any phase except Job Removal. After Job Removal, the Get-Job-Attributes and Get-Jobs operations no  
3325 longer are capable of returning any information about a job.

#### 3326 4.3.8 job-state-reasons (1setOf type2 keyword)

3327 This REQUIRED attribute provides additional information about the job's current state, i.e., information  
3328 that augments the value of the job's "job-state" attribute.

3329 These values MAY be used with any job state or states for which the reason makes sense. Some of these  
3330 value definitions indicate conformance requirements; the rest are OPTIONAL. Furthermore, when  
3331 implemented, the Printer MUST return these values when the reason applies and MUST NOT return them  
3332 when the reason no longer applies whether the value of the Job's "job-state" attribute changed or not. When

3333 the Job does not have any reasons for being in its current state, the value of the Job's "job-state-reasons"  
3334 attribute MUST be 'none'.

3335 Note: While values cannot be added to the 'job-state' attribute without impacting deployed clients that take  
3336 actions upon receiving "job-state" values, it is the intent that additional "job-state-reasons" values can be  
3337 defined and registered without impacting such deployed clients. In other words, the "job-state-reasons"  
3338 attribute is intended to be extensible.

3339 The following standard keyword values are defined. For ease of understanding, the values are presented in  
3340 the order in which the reasons are likely to occur (if implemented), starting with the 'job-incoming' value:

3341 'none': There are no reasons for the job's current state. This state reason is semantically equivalent to  
3342 "job-state-reasons" without any value and MUST be used when there is no other value, since the  
3343 1setOf attribute syntax requires at least one value.

3344 'job-incoming': The Create-Job operation has been accepted by the Printer, but the Printer is expecting  
3345 additional Send-Document and/or Send-URI operations and/or is accessing/accepting document  
3346 data.

3347 'job-data-insufficient': The Create-Job operation has been accepted by the Printer, but the Printer is  
3348 expecting additional document data before it can move the job into the 'processing' state. If a Printer  
3349 starts processing before it has received all data, the Printer removes the 'job-data-insufficient'  
3350 reason, but the 'job-incoming' remains. If a Printer starts processing after it has received all data, the  
3351 Printer removes the 'job-data-insufficient' reason and the 'job-incoming' at the same time.

3352 'document-access-error': After accepting a Print-URI or Send-URI request, the Printer could not access  
3353 one or more documents passed by reference. This reason is intended to cover any file access  
3354 problem, including file does not exist and access denied because of an access control problem. The  
3355 Printer MAY also indicate the document access error using the "job-document-access-errors" Job  
3356 Description attribute (see section 4.3.11). Whether the Printer aborts the job and moves the job to  
3357 the 'aborted' job state or prints all documents that are accessible and moves the job to the 'completed'  
3358 job state and adds the 'completed-with-errors' value in the job's "job-state-reasons" attribute depends  
3359 on implementation and/or site policy. This value SHOULD be supported if the Print-URI or Send-  
3360 URI operations are supported.

3361 'submission-interrupted': The job was not completely submitted for some unforeseen reason, such as:  
3362 (1) the Printer has crashed before the job was closed by the client, (2) the Printer or the document  
3363 transfer method has crashed in some non-recoverable way before the document data was entirely  
3364 transferred to the Printer, (3) the client crashed or failed to close the job before the time-out period.  
3365 See section 4.4.31.

3366 'job-outgoing': The Printer is transmitting the job to the output device.

3367 'job-hold-until-specified': The value of the job's "job-hold-until" attribute was specified with a time  
3368 period that is still in the future. The job MUST NOT be a candidate for processing until this reason  
3369 is removed and there are no other reasons to hold the job. This value SHOULD be supported if the  
3370 "job-hold-until" Job Template attribute is supported.

3371 'resources-are-not-ready': At least one of the resources needed by the job, such as media, fonts, resource  
3372 objects, etc., is not ready on any of the physical printer's for which the job is a candidate. This  
3373 condition MAY be detected when the job is accepted, or subsequently while the job is pending or  
3374 processing, depending on implementation. The job may remain in its current state or be moved to  
3375 the 'pending-held' state, depending on implementation and/or job scheduling policy.

- 3376 'printer-stopped-partly': The value of the Printer's "printer-state-reasons" attribute contains the value  
3377 'stopped-partly'.
- 3378 'printer-stopped': The value of the Printer's "printer-state" attribute is 'stopped'.
- 3379 'job-interpreting': Job is in the 'processing' state, but more specifically, the Printer is interpreting the  
3380 document data.
- 3381 'job-queued': Job is in the 'processing' state, but more specifically, the Printer has queued the document  
3382 data.
- 3383 'job-transforming': Job is in the 'processing' state, but more specifically, the Printer is interpreting  
3384 document data and producing another electronic representation.
- 3385 'job-queued-for-marker': Job is in any of the 'pending-held', 'pending', or 'processing' states, but more  
3386 specifically, the Printer has completed enough processing of the document to be able to start  
3387 marking and the job is waiting for the marker. Systems that require human intervention to release  
3388 jobs using the Release-Job operation, put the job into the 'pending-held' job state. Systems that  
3389 automatically select a job to use the marker put the job into the 'pending' job state or keep the job in  
3390 the 'processing' job state while waiting for the marker, depending on implementation. All  
3391 implementations put the job into (or back into) the 'processing' state when marking does begin.
- 3392 'job-printing': The output device is marking media. This value is useful for Printers which spend a great  
3393 deal of time processing (1) when no marking is happening and then want to show that marking is  
3394 now happening or (2) when the job is in the process of being canceled or aborted while the job  
3395 remains in the 'processing' state, but the marking has not yet stopped so that impression or sheet  
3396 counts are still increasing for the job.
- 3397 'job-canceled-by-user': The job was canceled by the owner of the job using the Cancel-Job request, i.e.,  
3398 by a user whose authenticated identity is the same as the value of the originating user that created  
3399 the Job object, or by some other authorized end-user, such as a member of the job owner's security  
3400 group. This value SHOULD be supported.
- 3401 'job-canceled-by-operator': The job was canceled by the operator using the Cancel-Job request, i.e., by a  
3402 user who has been authenticated as having operator privileges (whether local or remote). If the  
3403 security policy is to allow anyone to cancel anyone's job, then this value may be used when the job  
3404 is canceled by other than the owner of the job. For such a security policy, in effect, everyone is an  
3405 operator as far as canceling jobs with IPP is concerned. This value SHOULD be supported if the  
3406 implementation permits canceling by other than the owner of the job.
- 3407 'job-canceled-at-device': The job was canceled by an unidentified local user, i.e., a user at a console at  
3408 the device. This value SHOULD be supported if the implementation supports canceling jobs at the  
3409 console.
- 3410 'aborted-by-system': The job (1) is in the process of being aborted, (2) has been aborted by the system  
3411 and placed in the 'aborted' state, or (3) has been aborted by the system and placed in the 'pending-  
3412 held' state, so that a user or operator can manually try the job again. This value SHOULD be  
3413 supported.
- 3414 'unsupported-compression': The job was aborted by the system because the Printer determined while  
3415 attempting to decompress the document-data's that the compression is actually not among those  
3416 supported by the Printer. This value MUST be supported, since "compressions is a REQUIRED  
3417 operation attribute.
- 3418 'compression-error': The job was aborted by the system because the Printer encountered an error in the  
3419 document-data while decompressing it. If the Printer posts this reason, the document-data has  
3420 already passed any tests that would have led to the 'unsupported-compression' job-state-reason.

3421 'unsupported-document-format': The job was aborted by the system because the document-data's  
3422 document-format is not among those supported by the Printer. If the client specifies the document-  
3423 format as 'application/octet-stream', the printer MAY abort the job and post this reason even though  
3424 the format is a member of the "document-format-supported" printer attribute, but not among the  
3425 auto-sensed document-formats. This value MUST be supported, since "document-format" is a  
3426 REQUIRED operation attribute.

3427 'document-format-error': The job was aborted by the system because the Printer encountered an error in  
3428 the document-data while processing it. If the Printer posts this reason, the document-data has  
3429 already passed any tests that would have led to the 'unsupported-document-format' job-state-reason.

3430 'processing-to-stop-point': The requester has issued a Cancel-Job operation or the Printer object has  
3431 aborted the job, but is still performing some actions on the job until a specified stop point occurs or  
3432 job termination/cleanup is completed.

3433 If the implementation requires some measurable time to cancel the job in the 'processing' or  
3434 'processing-stopped' job states, the IPP object MUST use this value to indicate that the Printer object  
3435 is still performing some actions on the job while the job remains in the 'processing' or 'processing-  
3436 stopped' state. After all the job's job description attributes have stopped incrementing, the Printer  
3437 object moves the job from the 'processing' state to the 'canceled' or 'aborted' job states.

3438 'service-off-line': The Printer is off-line and accepting no jobs. All 'pending' jobs are put into the  
3439 'pending-held' state. This situation could be true if the service's or document transform's input is  
3440 impaired or broken.

3441 'job-completed-successfully': The job completed successfully. This value SHOULD be supported.

3442 'job-completed-with-warnings': The job completed with warnings. This value SHOULD be supported  
3443 if the implementation detects warnings.

3444 'job-completed-with-errors': The job completed with errors (and possibly warnings too). This value  
3445 SHOULD be supported if the implementation detects errors.

3446 'job-restartable' - This job is retained (see section 4.3.7.2) and is currently able to be restarted using the  
3447 Restart-Job operation (see section 3.3.7). If 'job-restartable' is a value of the job's 'job-state-reasons'  
3448 attribute, then the IPP object MUST accept a Restart-Job operation for that job. This value  
3449 SHOULD be supported if the Restart-Job operation is supported.

3450 'queued-in-device': The job has been forwarded to a device or print system that is unable to send back  
3451 status. The Printer sets the job's "job-state" attribute to 'completed' and adds the 'queued-in-device'  
3452 value to the job's "job-state-reasons" attribute to indicate that the Printer has no additional  
3453 information about the job and never will have any better information. See section 4.3.7.1.

#### 3454 4.3.9 job-state-message (text(MAX))

3455 This attribute specifies information about the "job-state" and "job-state-reasons" attributes in human  
3456 readable text. If the Printer object supports this attribute, the Printer object MUST be able to generate this  
3457 message in any of the natural languages identified by the Printer's "generated-natural-language-supported"  
3458 attribute (see the "attributes-natural-language" operation attribute specified in Section 3.1.4.1).

3459 The value SHOULD NOT contain additional information not contained in the values of the "job-state" and  
3460 "job-states-reasons" attributes, such as interpreter error information. Otherwise, application programs  
3461 might attempt to parse the (localized text). For such additional information such as interpreter errors for

3462 application program consumption or specific document access errors, new attributes with keyword values,  
3463 needs to be developed and registered.

#### 3464 4.3.10 job-detailed-status-messages (1setOf text(MAX))

3465 This attribute specifies additional detailed and technical information about the job. Neither the Printer nor  
3466 the client localizes the message(s), since they are intended for use by the system administrator or other  
3467 experienced technical persons. Clients **MUST NOT** attempt to parse the value of this attribute. See "job-  
3468 document-access-errors" (section 4.3.11) for additional errors that a program can process.

#### 3469 4.3.11 job-document-access-errors (1setOf text(MAX))

3470 This attribute provides additional information about each document access error for this job encountered by  
3471 the Printer after it returned a response to the Print-URI or Send-URI operation and subsequently attempted  
3472 to access document(s) supplied in the Print-URI or Send-URI operation. For errors in the protocol that is  
3473 identified by the URI scheme in the "document-uri" operation attribute, such as 'http:' or 'ftp:', the error code  
3474 is returned in parentheses, followed by the URI. For example:

3475 (404) http://ftp.pwg.org/pub/pwg/ipp/new\_MOD/ipp-model-v11-990510.pdf  
3476

3477 Most Internet protocols use decimal error codes (unlike IPP), so the ASCII error code representation is in  
3478 decimal.

#### 3479 4.3.12 number-of-documents (integer(0:MAX))

3480 This attribute indicates the number of documents in the job, i.e., the number of Send-Document, Send-URI,  
3481 Print-Job, or Print-URI operations that the Printer has accepted for this job, regardless of whether the  
3482 document data has reached the Printer object or not.

3483 Implementations supporting the OPTIONAL Create-Job/Send-Document/Send-URI operations **SHOULD**  
3484 support this attribute so that clients can query the number of documents in each job.

#### 3485 4.3.13 output-device-assigned (name(127))

3486 This attribute identifies the output device to which the Printer object has assigned this job. If an output  
3487 device implements an embedded Printer object, the Printer object **NEED NOT** set this attribute. If a print  
3488 server implements a Printer object, the value **MAY** be empty (zero-length string) or not returned until the  
3489 Printer object assigns an output device to the job. This attribute is particularly useful when a single Printer  
3490 object support multiple devices (so called "fan-out").

#### 3491 4.3.14 Event Time Job Description Attributes

3492 This section defines the Job Description attributes that indicate the time at which certain events occur for a  
3493 job. If the job event has not yet occurred, then the IPP object **MUST** return the 'no-value' out-of-band value  
3494 (see the beginning of Section 4.1). The "time-at-xxx(integer)" attributes represent time as an 'integer'  
3495 representing the number of seconds since the device was powered up (informally called "time ticks"). The

3496 "date-time-at-xxx(dateTime)" attributes represent time as 'dateTime' representing date and time (including  
3497 an offset from UTC).

3498 In order to populate these attributes, the Printer object copies the value(s) of the following Printer  
3499 Description attributes at the time the event occurs:

- 3500 1. the value in the Printer's "printer-up-time" attribute for the "time-at-xxx(integer)" attributes
- 3501 2. the value in the Printer's "printer-current-time" attribute for the "date-time-at-xxx(dateTime)"  
3502 attributes.

3503 If the Printer resets its "printer-up-time" attribute to 1 on power-up (see section 4.4.29) and has persistent  
3504 jobs, then it MUST change all of jobs' "time-at-xxx(integer)" (time tick) job attributes whose events have  
3505 occurred either to:

- 3506 1. 0 to indicate that the event happened before the most recent power up OR
- 3507 2. the negative of the number of seconds before the most recent power-up that the event took place,  
3508 though the negative number NEED NOT reflect the exact number of seconds.

3509 If a client queries a "time-at-xxx(integer)" time tick Job attribute and finds the value to be 0 or negative, the  
3510 client MUST assume that the event occurred in some life other than the Printer's current life.

3511 Note: A Printer does not change the values of any "date-time-at-xxx(dateTime)" job attributes on power-up.

#### 3512 4.3.14.1 time-at-creation (integer(MIN:MAX))

3513 This REQUIRED attribute indicates the time at which the Job object was created.

#### 3514 4.3.14.2 time-at-processing (integer(MIN:MAX))

3515 This REQUIRED attribute indicates the time at which the Job object first began processing after the create  
3516 operation or the most recent Restart-Job operation. The out-of-band 'no-value' value is returned if the job  
3517 has not yet been in the 'processing' state (see the beginning of Section 4.1).

#### 3518 4.3.14.3 time-at-completed (integer(MIN:MAX))

3519 This REQUIRED attribute indicates the time at which the Job object completed (or was cancelled or  
3520 aborted). The out-of-band 'no-value' value is returned if the job has not yet completed, been canceled, or  
3521 aborted (see the beginning of Section 4.1).

#### 3522 4.3.14.4 job-printer-up-time (integer(1:MAX))

3523 This REQUIRED Job Description attribute indicates the amount of time (in seconds) that the Printer  
3524 implementation has been up and running. This attribute is an alias for the "printer-up-time" Printer  
3525 Description attribute (see Section 4.4.29).

3526 A client MAY request this attribute in a Get-Job-Attributes or Get-Jobs request and use the value returned  
3527 in combination with other requested Event Time Job Description Attributes in order to display time  
3528 attributes to a user. The difference between this attribute and the "integer" value of a "time-at-xxx" attribute  
3529 is the number of seconds ago that the "time-at-xxx" event occurred. A client can compute the wall-clock  
3530 time at which the "time-at-xxx" event occurred by subtracting this difference from the client's wall-clock  
3531 time.

#### 3532 4.3.14.5 date-time-at-creation (dateTime)

3533 This attribute indicates the date and time at which the Job object was created.

#### 3534 4.3.14.6 date-time-at-processing (dateTime)

3535 This attribute indicates the date and time at which the Job object first began processing after the create  
3536 operation or the most recent Restart-Job operation.

#### 3537 4.3.14.7 date-time-at-completed (dateTime)

3538 This attribute indicates the date and time at which the Job object completed (or was cancelled or aborted).

#### 3539 4.3.15 number-of-intervening-jobs (integer(0:MAX))

3540 This attribute indicates the number of jobs that are "ahead" of this job in the relative chronological order of  
3541 expected time to complete (i.e., the current scheduled order). For efficiency, it is only necessary to calculate  
3542 this value when an operation is performed that requests this attribute.

#### 3543 4.3.16 job-message-from-operator (text(127))

3544 This attribute provides a message from an operator, system administrator or "intelligent" process to indicate  
3545 to the end user the reasons for modification or other management action taken on a job.

#### 3546 4.3.17 Job Size Attributes

3547 This sub-section defines job attributes that describe the size of the job. These attributes are not intended to  
3548 be counters; they are intended to be useful routing and scheduling information if known. For these  
3549 attributes, the Printer object may try to compute the value if it is not supplied in the create request. Even if  
3550 the client does supply a value for these three attributes in the create request, the Printer object MAY choose  
3551 to change the value if the Printer object is able to compute a value which is more accurate than the client  
3552 supplied value. The Printer object may be able to determine the correct value for these attributes either  
3553 right at job submission time or at any later point in time.



## 3554 4.3.17.1 job-k-octets (integer(0:MAX))

3555 This attribute specifies the total size of the document(s) in K octets, i.e., in units of 1024 octets requested to  
3556 be processed in the job. The value MUST be rounded up, so that a job between 1 and 1024 octets MUST  
3557 be indicated as being 1, 1025 to 2048 MUST be 2, etc.

3558 This value MUST NOT include the multiplicative factors contributed by the number of copies specified by  
3559 the "copies" attribute, independent of whether the device can process multiple copies without making  
3560 multiple passes over the job or document data and independent of whether the output is collated or not.  
3561 Thus the value is independent of the implementation and indicates the size of the document(s) measured in  
3562 K octets independent of the number of copies.

3563 This value MUST also not include the multiplicative factor due to a copies instruction embedded in the  
3564 document data. If the document data actually includes replications of the document data, this value will  
3565 include such replication. In other words, this value is always the size of the source document data, rather  
3566 than a measure of the hardcopy output to be produced.

## 3567 4.3.17.2 job-impressions (integer(0:MAX))

3568 This attribute specifies the total size in number of impressions of the document(s) being submitted (see the  
3569 definition of impression in section 12.2.5).

3570 As with "job-k-octets", this value MUST NOT include the multiplicative factors contributed by the number  
3571 of copies specified by the "copies" attribute, independent of whether the device can process multiple copies  
3572 without making multiple passes over the job or document data and independent of whether the output is  
3573 collated or not. Thus the value is independent of the implementation and reflects the size of the  
3574 document(s) measured in impressions independent of the number of copies.

3575 As with "job-k-octets", this value MUST also not include the multiplicative factor due to a copies  
3576 instruction embedded in the document data. If the document data actually includes replications of the  
3577 document data, this value will include such replication. In other words, this value is always the number of  
3578 impressions in the source document data, rather than a measure of the number of impressions to be  
3579 produced by the job.

## 3580 4.3.17.3 job-media-sheets (integer(0:MAX))

3581 This attribute specifies the total number of media sheets to be produced for this job.

3582 Unlike the "job-k-octets" and the "job-impressions" attributes, this value MUST include the multiplicative  
3583 factors contributed by the number of copies specified by the "copies" attribute and a 'number of copies'  
3584 instruction embedded in the document data, if any. This difference allows the system administrator to  
3585 control the lower and upper bounds of both (1) the size of the document(s) with "job-k-octets-supported"  
3586 and "job-impressions-supported" and (2) the size of the job with "job-media-sheets-supported".

## 3587 4.3.18 Job Progress Attributes

3588 This sub-section defines job attributes that describe the progress of the job. These attributes are intended to  
3589 be counters. That is, the value for a job that has not started processing **MUST** be 0. When the job's "job-  
3590 state" is 'processing' or 'processing-stopped', this value is intended to contain the amount of the job that has  
3591 been processed to the time at which the attributes are requested. When the job enters the 'completed',  
3592 'canceled', or 'aborted' states, these values are the final values for the job.

## 3593 4.3.18.1 job-k-octets-processed (integer(0:MAX))

3594 This attribute specifies the total number of octets processed in K octets, i.e., in units of 1024 octets so far.  
3595 The value **MUST** be rounded up, so that a job between 1 and 1024 octets inclusive **MUST** be indicated as  
3596 being 1, 1025 to 2048 inclusive **MUST** be 2, etc.

3597 For implementations where multiple copies are produced by the interpreter with only a single pass over the  
3598 data, the final value **MUST** be equal to the value of the "job-k-octets" attribute. For implementations where  
3599 multiple copies are produced by the interpreter by processing the data for each copy, the final value **MUST**  
3600 be a multiple of the value of the "job-k-octets" attribute.

## 3601 4.3.18.2 job-impressions-completed (integer(0:MAX))

3602 This job attribute specifies the number of impressions completed for the job so far. For printing devices,  
3603 the impressions completed includes interpreting, marking, and stacking the output.

## 3604 4.3.18.3 job-media-sheets-completed (integer(0:MAX))

3605 This job attribute specifies the media-sheets completed marking and stacking for the entire job so far  
3606 whether those sheets have been processed on one side or on both.

## 3607 4.3.19 attributes-charset (charset)

3608 This **REQUIRED** attribute is populated using the value in the client supplied "attributes-charset" attribute in  
3609 the create request. It identifies the charset (coded character set and encoding method) used by any Job  
3610 attributes with attribute syntax 'text' and 'name' that were supplied by the client in the create request. See  
3611 Section 3.1.4 for a complete description of the "attributes-charset" operation attribute.

3612 This attribute does not indicate the charset in which the 'text' and 'name' values are stored internally in the  
3613 Job object. The internal charset is implementation-defined. The IPP object **MUST** convert from whatever  
3614 the internal charset is to that being requested in an operation as specified in Section 3.1.4.

## 3615 4.3.20 attributes-natural-language (naturalLanguage)

3616 This **REQUIRED** attribute is populated using the value in the client supplied "attributes-natural-language"  
3617 attribute in the create request. It identifies the natural language used for any Job attributes with attribute  
3618 syntax 'text' and 'name' that were supplied by the client in the create request. See Section 3.1.4 for a

3619 complete description of the "attributes-natural-language" operation attribute. See Sections 4.1.1.2 and  
3620 4.1.2.2 for how a Natural Language Override may be supplied explicitly for each 'text' and 'name' attribute  
3621 value that differs from the value identified by the "attributes-natural-language" attribute.

#### 3622 4.4 Printer Description Attributes

3623 These attributes form the attribute group called "printer-description". The following table summarizes  
3624 these attributes, their syntax, and whether or not they are REQUIRED for a Printer object to support. If  
3625 they are not indicated as REQUIRED, they are OPTIONAL. The maximum size in octets for 'text' and  
3626 'name' attributes is indicated in parentheses.

3627 Note: How these attributes are set by an Administrator is outside the scope of this IPP/1.1 document.

3628	+-----+-----+-----+
3629	Attribute   Syntax   REQUIRED?
3630	+-----+-----+-----+
3631	printer-uri-supported   1setOf uri   REQUIRED
3632	+-----+-----+-----+
3633	uri-security-supported   1setOf type2 keyword   REQUIRED
3634	+-----+-----+-----+
3635	uri-authentication- supported   1setOf type2 keyword   REQUIRED
3636	+-----+-----+-----+
3637	+-----+-----+-----+
3638	printer-name   name (127)   REQUIRED
3639	+-----+-----+-----+
3640	printer-location   text (127)
3641	+-----+-----+-----+
3642	printer-info   text (127)
3643	+-----+-----+-----+
3644	printer-more-info   uri
3645	+-----+-----+-----+
3646	printer-driver-installer   uri
3647	+-----+-----+-----+
3648	printer-make-and-model   text (127)
3649	+-----+-----+-----+
3650	printer-more-info- manufacturer   uri
3651	+-----+-----+-----+
3652	+-----+-----+-----+
3653	printer-state   type1 enum   REQUIRED
3654	+-----+-----+-----+
3655	printer-state-reasons   1setOf type2 keyword   REQUIRED
3656	+-----+-----+-----+
3657	printer-state-message   text (MAX)
3658	+-----+-----+-----+
3659	ipp-versions-supported   1setOf type2 keyword   REQUIRED
3660	+-----+-----+-----+
3661	operations-supported   1setOf type2 enum   REQUIRED
3662	+-----+-----+-----+
3663	ipp-multiple-document-jobs- supported   boolean
3664	+-----+-----+-----+
3665	+-----+-----+-----+
3666	charset-configured   charset   REQUIRED
3667	+-----+-----+-----+
3668	charset-supported   1setOf charset   REQUIRED
3669	+-----+-----+-----+
3670	natural-language-configured   naturalLanguage   REQUIRED
3671	+-----+-----+-----+
3672	generated-natural-language- supported   1setOf naturalLanguage   REQUIRED
3673	+-----+-----+-----+
3674	+-----+-----+-----+
3675	document-format-default   mimeType   REQUIRED
3676	+-----+-----+-----+
3677	document-format-supported   1setOf mimeType   REQUIRED

3678	+-----+-----+-----+
3679	printer-is-accepting-jobs   boolean   REQUIRED
3680	+-----+-----+-----+
3681	queued-job-count   integer (0:MAX)   REQUIRED
3682	+-----+-----+-----+
3683	printer-message-from-
3684	operator   text (127)
3685	+-----+-----+-----+
3686	color-supported   boolean
3687	+-----+-----+-----+
3688	reference-uri-schemes-
3689	supported   1setOf uriScheme
3690	+-----+-----+-----+
3691	pdl-override-supported   type2 keyword   REQUIRED
3692	+-----+-----+-----+
3693	printer-up-time   integer (1:MAX)   REQUIRED
3694	+-----+-----+-----+
3695	printer-current-time   dateTime
3696	+-----+-----+-----+
3697	multiple-operation-time-out   integer (1:MAX)
3698	+-----+-----+-----+
3699	compression-supported   1setOf type3 keyword   REQUIRED
3700	+-----+-----+-----+
3701	job-k-octets-supported   rangeOfInteger (0:MAX)
3702	+-----+-----+-----+
3703	job-impressions-supported   rangeOfInteger (0:MAX)
3704	+-----+-----+-----+
3705	job-media-sheets-supported   rangeOfInteger (0:MAX)
3706	+-----+-----+-----+
3707	pages-per-minute   integer(0:MAX)
3708	+-----+-----+-----+
3709	pages-per-minute-color   integer(0:MAX)
3710	+-----+-----+-----+
3711	

#### 3712 4.4.1 printer-uri-supported (1setOf uri)

3713 This REQUIRED Printer attribute contains at least one URI for the Printer object. It OPTIONALLY  
 3714 contains more than one URI for the Printer object. An administrator determines a Printer object's URI(s)  
 3715 and configures this attribute to contain those URIs by some means outside the scope of this IPP/1.1  
 3716 document. The precise format of this URI is implementation dependent and depends on the protocol. See  
 3717 the next two sections for a description of the "uri-security-supported" and "uri-authentication-supported"  
 3718 attributes, both of which are the REQUIRED companion attributes to this "printer-uri-supported" attribute.  
 3719 See section 2.4 on Printer object identity and section 8.2 on security and URIs for more information.

#### 3720 4.4.2 uri-authentication-supported (1setOf type2 keyword)

3721 This REQUIRED Printer attribute MUST have the same cardinality (contain the same number of values) as  
 3722 the "printer-uri-supported" attribute. This attribute identifies the Client Authentication mechanism

3723 associated with each URI listed in the "printer-uri-supported" attribute. The Printer object uses the specified  
3724 mechanism to identify the authenticated user (see section 8.3) . The "i th" value in "uri-authentication-  
3725 supported" corresponds to the "i th" value in "printer-uri-supported" and it describes the authentication  
3726 mechanisms used by the Printer when accessed via that URI. See [IPP-PRO] for more details on Client  
3727 Authentication.

3728 The following standard keyword values are defined:

3729 'none': There is no authentication mechanism associated with the URI. The Printer object assumes that  
3730 the authenticated user is "anonymous".

3731 'requesting-user-name': When a client performs an operation whose target is the associated URI, the  
3732 Printer object assumes that the authenticated user is specified by the "requesting-user-name"  
3733 Operation attribute (see section 8.3). If the "requesting-user-name" attribute is absent in a request,  
3734 the Printer object assumes that the authenticated user is "anonymous".

3735 'basic': When a client performs an operation whose target is the associated URI, the Printer object  
3736 challenges the client with HTTP basic authentication. The Printer object assumes that the  
3737 authenticated user is the name received via the basic authentication mechanism.

3738 'digest': When a client performs an operation whose target is the associated URI, the Printer object  
3739 challenges the client with HTTP digest authentication. The Printer object assumes that the  
3740 authenticated user is the name received via the digest authentication mechanism.

3741 'certificate': When a client performs an operation whose target is the associated URI, the Printer object  
3742 expects the client to provide a certificate. The Printer object assumes that the authenticated user is  
3743 the textual name contained within the certificate.

#### 3744 4.4.3 uri-security-supported (1setOf type2 keyword)

3745 This REQUIRED Printer attribute MUST have the same cardinality (contain the same number of values) as  
3746 the "printer-uri-supported" attribute. This attribute identifies the security mechanisms used for each URI  
3747 listed in the "printer-uri-supported" attribute. The "i th" value in "uri-security-supported" corresponds to  
3748 the "i th" value in "printer-uri-supported" and it describes the security mechanisms used for accessing the  
3749 Printer object via that URI. See [IPP-PRO] for more details on security mechanisms.

3750 The following standard keyword values are defined:

3751 'none': There are no secure communication channel protocols in use for the given URI.

3752 'ssl3': SSL3 [SSL] is the secure communications channel protocol in use for the given URI.

3753 'tls': TLS [RFC2246] is the secure communications channel protocol in use for the given URI.  
3754

3755 This attribute is orthogonal to the definition of a Client Authentication mechanism. Specifically, 'none'  
3756 does not exclude Client Authentication. See section 4.4.2.

3757 Consider the following example. For a single Printer object, an administrator configures the "printer-uri-  
3758 supported", "uri-authentication-supported" and "uri-security-supported" attributes as follows:

3759 "printer-uri-supported": 'xxx://acme.com/open-use-printer', 'xxx://acme.com/restricted-use-printer',  
3760 'xxx://acme.com/private-printer'

3761 "uri-authentication-supported": 'none', 'digest', 'basic'

3762 "uri-security-supported": 'none', 'none', 'tls'  
3763

3764 Note: 'xxx' is not a valid scheme. See the IPP/1.1 "Transport and Encoding" document [IPP-PRO] for the  
3765 actual URI schemes to be used in object target attributes.

3766 In this case, one Printer object has three URIs.

- 3767 - For the first URI, 'xxx://acme.com/open-use-printer', the value 'none' in "uri-security-supported"  
3768 indicates that there is no secure channel protocol configured to run under HTTP. The value of 'none'  
3769 in "uri-authentication-supported" indicates that all users are 'anonymous'. There will be no  
3770 challenge and the Printer will ignore "requesting-user-name".
- 3771 - For the second URI, 'xxx://acme.com/restricted-use-printer', the value 'none' in "uri-security-  
3772 supported" indicates that there is no secure channel protocol configured to run under HTTP. The  
3773 value of 'digest' in "uri-authentication-supported" indicates that the Printer will issue a challenge and  
3774 that the Printer will use the name supplied by the digest mechanism to determine the authenticated  
3775 user (see section 8.3).
- 3776 - For the third URI, 'xxx://acme.com/private-printer', the value 'tls' in "uri-security-supported" indicates  
3777 that TLS is being used to secure the channel. The client SHOULD be prepared to use TLS framing  
3778 to negotiate an acceptable ciphersuite to use while communicating with the Printer object. In this  
3779 case, the name implies the use of a secure communications channel, but the fact is made explicit by  
3780 the presence of the 'tls' value in "uri-security-supported". The client does not need to resort to  
3781 understanding which security it must use by following naming conventions or by parsing the URI to  
3782 determine which security mechanisms are implied. The value of 'basic' in "uri-authentication-  
3783 supported" indicates that the Printer will issue a challenge and that the Printer will use the name  
3784 supplied by the digest mechanism to determine the authenticated user (see section 8.3) . Because  
3785 this challenge occurs in a tls session, the channel is secure.  
3786

3787 It is expected that many IPP Printer objects will be configured to support only one channel (either  
3788 configured to use TLS access or not) and only one authentication mechanism. Such Printer objects only  
3789 have one URI listed in the "printer-uri-supported" attribute. No matter the configuration of the Printer  
3790 object (whether it has only one URI or more than one URI), a client MUST supply only one URI in the  
3791 target "printer-uri" operation attribute.

#### 3792 4.4.4 printer-name (name(127))

3793 This REQUIRED Printer attribute contains the name of the Printer object. It is a name that is more end-  
3794 user friendly than a URI. An administrator determines a printer's name and sets this attribute to that name.  
3795 This name may be the last part of the printer's URI or it may be unrelated. In non-US-English locales, a  
3796 name may contain characters that are not allowed in a URI.

#### 3797 4.4.5 printer-location (text(127))

3798 This Printer attribute identifies the location of the device. This could include things like: "in Room 123A,  
3799 second floor of building XYZ".

## 3800 4.4.6 printer-info (text(127))

3801 This Printer attribute identifies the descriptive information about this Printer object. This could include  
3802 things like: "This printer can be used for printing color transparencies for HR presentations", or "Out of  
3803 courtesy for others, please print only small (1-5 page) jobs at this printer", or even "This printer is going  
3804 away on July 1, 1997, please find a new printer".

## 3805 4.4.7 printer-more-info (uri)

3806 This Printer attribute contains a URI used to obtain more information about this specific Printer object. For  
3807 example, this could be an HTTP type URI referencing an HTML page accessible to a Web Browser. The  
3808 information obtained from this URI is intended for end user consumption. Features outside the scope of IPP  
3809 can be accessed from this URI. The information is intended to be specific to this printer instance and site  
3810 specific services (e.g. job pricing, services offered, end user assistance). The device manufacturer may  
3811 initially populate this attribute.

## 3812 4.4.8 printer-driver-installer (uri)

3813 This Printer attribute contains a URI to use to locate the driver installer for this Printer object. This  
3814 attribute is intended for consumption by automata. The mechanics of print driver installation is outside the  
3815 scope of this IPP/1.1 document. The device manufacturer may initially populate this attribute.

## 3816 4.4.9 printer-make-and-model (text(127))

3817 This Printer attribute identifies the make and model of the device. The device manufacturer may initially  
3818 populate this attribute.

## 3819 4.4.10 printer-more-info-manufacturer (uri)

3820 This Printer attribute contains a URI used to obtain more information about this type of device. The  
3821 information obtained from this URI is intended for end user consumption. Features outside the scope of  
3822 IPP can be accessed from this URI (e.g., latest firmware, upgrades, print drivers, optional features available,  
3823 details on color support). The information is intended to be germane to this printer without regard to site  
3824 specific modifications or services. The device manufacturer may initially populate this attribute.

## 3825 4.4.11 printer-state (type1 enum)

3826 This REQUIRED Printer attribute identifies the current state of the device. The "printer-state reasons"  
3827 attribute augments the "printer-state" attribute to give more detailed information about the Printer in the  
3828 given printer state.

3829 A Printer object need only update this attribute before responding to an operation which requests the  
3830 attribute; the Printer object NEED NOT update this attribute continually, since asynchronous event  
3831 notification is not part of IPP/1.1. A Printer NEED NOT implement all values if they are not applicable to  
3832 a given implementation.



3833 The following standard enum values are defined:

3834 Value Symbolic Name and Description

3835

3836 '3' 'idle': Indicates that new jobs can start processing without waiting.

3837 '4' 'processing': Indicates that jobs are processing; new jobs will wait before processing.

3838 '5' 'stopped': Indicates that no jobs can be processed and intervention is required.

3839 Values of "printer-state-reasons", such as 'spool-area-full' and 'stopped-partly', MAY be used to provide  
3840 further information.

3841 4.4.12 printer-state-reasons (1setOf type2 keyword)

3842 This REQUIRED Printer attribute supplies additional detail about the device's state. Some of the these  
3843 value definitions indicate conformance requirements; the rest are OPTIONAL.

3844 Each keyword value MAY have a suffix to indicate its level of severity. The three levels are: report (least  
3845 severe), warning, and error (most severe).

3846 - '-report': This suffix indicates that the reason is a "report". An implementation may choose to omit  
3847 some or all reports. Some reports specify finer granularity about the printer state; others serve as a  
3848 precursor to a warning. A report MUST contain nothing that could affect the printed output.

3849 - '-warning': This suffix indicates that the reason is a "warning". An implementation may choose to omit  
3850 some or all warnings. Warnings serve as a precursor to an error. A warning MUST contain nothing  
3851 that prevents a job from completing, though in some cases the output may be of lower quality.

3852 - '-error': This suffix indicates that the reason is an "error". An implementation MUST include all  
3853 errors. If this attribute contains one or more errors, printer MUST be in the stopped state.  
3854

3855 If the implementation does not add any one of the three suffixes, all parties MUST assume that the reason is  
3856 an "error".

3857 If a Printer object controls more than one output device, each value of this attribute MAY apply to one or  
3858 more of the output devices. An error on one output device that does not stop the Printer object as a whole  
3859 MAY appear as a warning in the Printer's "printer-state-reasons attribute". If the "printer-state" for such a  
3860 Printer has a value of 'stopped', then there MUST be an error reason among the values in the "printer-state-  
3861 reasons" attribute.

3862 The following standard keyword values are defined:

3863 'other': The device has detected an error other than one listed in this document.

3864 'none': There are not reasons. This state reason is semantically equivalent to "printer-state-reasons"  
3865 without any value and MUST be used, since the 1setOf attribute syntax requires at least one value.

3866 'media-needed': A tray has run out of media.

3867 'media-jam': The device has a media jam.

3868 'moving-to-paused': Someone has paused the Printer object using the Pause-Printer operation (see  
3869 section 3.2.7) or other means, but the device(s) are taking an appreciable time to stop. Later, when  
3870 all output has stopped, the "printer-state" becomes 'stopped', and the 'paused' value replaces the

3871 'moving-to-paused' value in the "printer-state-reasons" attribute. This value MUST be supported, if  
3872 the Pause-Printer operation is supported and the implementation takes significant time to pause a  
3873 device in certain circumstances.

3874 'paused': Someone has paused the Printer object using the Pause-Printer operation (see section 3.2.7) or  
3875 other means and the Printer object's "printer-state" is 'stopped'. In this state, a Printer MUST NOT  
3876 produce printed output, but it MUST perform other operations requested by a client. If a Printer had  
3877 been printing a job when the Printer was paused, the Printer MUST resume printing that job when  
3878 the Printer is no longer paused and leave no evidence in the printed output of such a pause. This  
3879 value MUST be supported, if the Pause-Printer operation is supported.

3880 'shutdown': Someone has removed a Printer object from service, and the device may be powered down  
3881 or physically removed. In this state, a Printer object MUST NOT produce printed output, and unless  
3882 the Printer object is realized by a print server that is still active, the Printer object MUST perform no  
3883 other operations requested by a client, including returning this value. If a Printer object had been  
3884 printing a job when it was shutdown, the Printer NEED NOT resume printing that job when the  
3885 Printer is no longer shutdown. If the Printer resumes printing such a job, it may leave evidence in  
3886 the printed output of such a shutdown, e.g. the part printed before the shutdown may be printed a  
3887 second time after the shutdown.

3888 'connecting-to-device': The Printer object has scheduled a job on the output device and is in the process  
3889 of connecting to a shared network output device (and might not be able to actually start printing the  
3890 job for an arbitrarily long time depending on the usage of the output device by other servers on the  
3891 network).

3892 'timed-out': The server was able to connect to the output device (or is always connected), but was unable  
3893 to get a response from the output device.

3894 'stopping': The Printer object is in the process of stopping the device and will be stopped in a while.  
3895 When the device is stopped, the Printer object will change the Printer object's state to 'stopped'. The  
3896 'stopping-warning' reason is never an error, even for a Printer with a single output device. When an  
3897 output-device ceases accepting jobs, the Printer will have this reason while the output device  
3898 completes printing.

3899 'stopped-partly': When a Printer object controls more than one output device, this reason indicates that  
3900 one or more output devices are stopped. If the reason is a report, fewer than half of the output  
3901 devices are stopped. If the reason is a warning, fewer than all of the output devices are stopped.

3902 'toner-low': The device is low on toner.

3903 'toner-empty': The device is out of toner.

3904 'spool-area-full': The limit of persistent storage allocated for spooling has been reached. The Printer is  
3905 temporarily unable to accept more jobs. The Printer will remove this value when it is able to accept  
3906 more jobs. This value SHOULD be used by a non-spooling Printer that only accepts one or a small  
3907 number jobs at a time or a spooling Printer that has filled the spool space.

3908 'cover-open': One or more covers on the device are open.

3909 'interlock-open': One or more interlock devices on the printer are unlocked.

3910 'door-open': One or more doors on the device are open.

3911 'input-tray-missing': One or more input trays are not in the device.

3912 'media-low': At least one input tray is low on media.

3913 'media-empty': At least one input tray is empty.

3914 'output-tray-missing': One or more output trays are not in the device

3915 'output-area-almost-full': One or more output area is almost full (e.g. tray, stacker, collator).

3916 'output-area-full': One or more output area is full. (e.g. tray, stacker, collator)  
3917 'marker-supply-low': The device is low on at least one marker supply. (e.g. toner, ink, ribbon)  
3918 'marker-supply-empty': The device is out of at least one marker supply. (e.g. toner, ink, ribbon)  
3919 'marker-waste-almost-full': The device marker supply waste receptacle is almost full.  
3920 'marker-waste-full': The device marker supply waste receptacle is full.  
3921 'fuser-over-temp': The fuser temperature is above normal.  
3922 'fuser-under-temp': The fuser temperature is below normal.  
3923 'opc-near-eol': The optical photo conductor is near end of life.  
3924 'opc-life-over': The optical photo conductor is no longer functioning.  
3925 'developer-low': The device is low on developer.  
3926 'developer-empty': The device is out of developer.  
3927 'interpreter-resource-unavailable': An interpreter resource is unavailable (i.e. font, form)  
3928

#### 3929 4.4.13 printer-state-message (text(MAX))

3930 This Printer attribute specifies the additional information about the printer state and printer state reasons in  
3931 human readable text. If the Printer object supports this attribute, the Printer object MUST be able to  
3932 generate this message in any of the natural languages identified by the Printer's "generated-natural-  
3933 language-supported" attribute (see the "attributes-natural-language" operation attribute specified in Section  
3934 3.1.4.1).

#### 3935 4.4.14 ipp-versions-supported (1setOf type2 keyword)

3936 This REQUIRED attribute identifies the IPP protocol version(s) that this Printer supports, including major  
3937 and minor versions, i.e., the version numbers for which this Printer implementation meets the conformance  
3938 requirements. For version number validation, the Printer matches the (two-octet binary) "version-number"  
3939 parameter supplied by the client in each request (see sections 3.1.1 and 3.1.8) with the (US-ASCII) keyword  
3940 values of this attribute.

3941 The following standard keyword values are defined:

3942 '1.0': Meets the conformance requirement of IPP version 1.0 as specified in RFC 2566 [RFC2566] and  
3943 RFC 2565 [RFC2565] including any extensions registered according to Section 6 and any extension  
3944 defined in this version or any future version of the IPP "Model and Semantics" document or the IPP  
3945 "Encoding and Transport" document following the rules, if any, when the "version-number"  
3946 parameter is '1.0'.

3947 '1.1': Meets the conformance requirement of IPP version 1.1 as specified in this document and [IPP-  
3948 PRO] including any extensions registered according to Section 6 and any extension defined in any  
3949 future versions of the IPP "Model and Semantics" document or the IPP Encoding and Transport  
3950 document following the rules, if any, when the "version-number" parameter is '1.1'.

#### 3951 4.4.15 operations-supported (1setOf type2 enum)

3952 This REQUIRED Printer attribute specifies the set of supported operations for this Printer object and  
3953 contained Job objects.

3954 This attribute is encoded as any other enum attribute syntax according to [IPP-PRO] as 32-bits. However,  
 3955 all 32-bit enum values for this attribute MUST NOT exceed 0x00008FFF, since these same values are also  
 3956 passed in two octets in the "operation-id" parameter (see section 3.1.1) in each Protocol request with the  
 3957 two high order octets omitted in order to indicate the operation being performed [IPP-PRO].

3958 The following standard enum and "operation-id" (see section 3.1.2) values are defined:

3959	Value	Operation Name
3960	-----	-----
3961		
3962	0x0000	reserved, not used
3963	0x0001	reserved, not used
3964	0x0002	Print-Job
3965	0x0003	Print-URI
3966	0x0004	Validate-Job
3967	0x0005	Create-Job
3968	0x0006	Send-Document
3969	0x0007	Send-URI
3970	0x0008	Cancel-Job
3971	0x0009	Get-Job-Attributes
3972	0x000A	Get-Jobs
3973	0x000B	Get-Printer-Attributes
3974	0x000C	Hold-Job
3975	0x000D	Release-Job
3976	0x000E	Restart-Job
3977	0x000F	reserved for a future operation
3978	0x0010	Pause-Printer
3979	0x0011	Resume-Printer
3980	0x0012	Purge-Jobs
3981	0x0013-0x3FFF	reserved for future operations
3982	0x4000-0x8FFF	reserved for private extensions
3983		

3984 The reserved block for private extensions allows for vendors to implement private extensions that are  
 3985 guaranteed to not conflict with future registered extensions. However, there is no guarantee that two or  
 3986 more private extensions will not conflict.

#### 3987 4.4.16 multiple-document-jobs-supported (boolean)

3988 This Printer attribute indicates whether or not the Printer supports more than one document per job, i.e.,  
 3989 more than one Send-Document or Send-Data operation with document data. If the Printer supports the  
 3990 Create-Job and Send-Document operations (see section 3.2.4 and 3.3.1), it MUST support this attribute.

## 3991 4.4.17 charset-configured (charset)

3992 This REQUIRED Printer attribute identifies the charset that the Printer object has been configured to  
3993 represent 'text' and 'name' Printer attributes that are set by the operator, system administrator, or  
3994 manufacturer, i.e., for "printer-name" (name), "printer-location" (text), "printer-info" (text), and "printer-  
3995 make-and-model" (text). Therefore, the value of the Printer object's "charset-configured" attribute MUST  
3996 also be among the values of the Printer object's "charset-supported" attribute.

## 3997 4.4.18 charset-supported (1setOf charset)

3998 This REQUIRED Printer attribute identifies the set of charsets that the Printer and contained Job objects  
3999 support in attributes with attribute syntax 'text' and 'name'. At least the value 'utf-8' MUST be present, since  
4000 IPP objects MUST support the UTF-8 [RFC2279] charset. If a Printer object supports a charset, it means  
4001 that for all attributes of syntax 'text' and 'name' the IPP object MUST (1) accept the charset in requests and  
4002 return the charset in responses as needed.

4003 If more charsets than UTF-8 are supported, the IPP object MUST perform charset conversion between the  
4004 charsets as described in Section 3.1.4.2.

## 4005 4.4.19 natural-language-configured (naturalLanguage)

4006 This REQUIRED Printer attribute identifies the natural language that the Printer object has been configured  
4007 to represent 'text' and 'name' Printer attributes that are set by the operator, system administrator, or  
4008 manufacturer, i.e., for "printer-name" (name), "printer-location" (text), "printer-info" (text), and "printer-  
4009 make-and-model" (text). When returning these Printer attributes, the Printer object MAY return them in the  
4010 configured natural language specified by this attribute, instead of the natural language requested by the  
4011 client in the "attributes-natural-language" operation attribute. See Section 3.1.4.1 for the specification of  
4012 the OPTIONAL multiple natural language support. Therefore, the value of the Printer object's "natural-  
4013 language-configured" attribute MUST also be among the values of the Printer object's "generated-natural-  
4014 language-supported" attribute.

## 4015 4.4.20 generated-natural-language-supported (1setOf naturalLanguage)

4016 This REQUIRED Printer attribute identifies the natural language(s) that the Printer object and contained  
4017 Job objects support in attributes with attribute syntax 'text' and 'name'. The natural language(s) supported  
4018 depends on implementation and/or configuration. Unlike charsets, IPP objects MUST accept requests with  
4019 any natural language or any Natural Language Override whether the natural language is supported or not.

4020 If a Printer object supports a natural language, it means that for any of the attributes for which the Printer or  
4021 Job object generates messages, i.e., for the "job-state-message" and "printer-state-message" attributes and  
4022 Operation Messages (see Section 3.1.5) in operation responses, the Printer and Job objects MUST be able  
4023 to generate messages in any of the Printer's supported natural languages. See section 3.1.4 for the definition  
4024 of 'text' and 'name' attributes in operation requests and responses.

4025 Note: A Printer object that supports multiple natural languages, often has separate catalogs of messages,  
4026 one for each natural language supported.

## 4027 4.4.21 document-format-default (mimeMediaType)

4028 This REQUIRED Printer attribute identifies the document format that the Printer object has been  
4029 configured to assume if the client does not supply a "document-format" operation attribute in any of the  
4030 operation requests that supply document data. The standard values for this attribute are Internet Media  
4031 types (sometimes called MIME types). For further details see the description of the 'mimeMediaType'  
4032 attribute syntax in Section 4.1.9.

## 4033 4.4.22 document-format-supported (1setOf mimeMediaType)

4034 This REQUIRED Printer attribute identifies the set of document formats that the Printer object and  
4035 contained Job objects can support. For further details see the description of the 'mimeMediaType' attribute  
4036 syntax in Section 4.1.9.

## 4037 4.4.23 printer-is-accepting-jobs (boolean)

4038 This REQUIRED Printer attribute indicates whether the printer is currently able to accept jobs, i.e., is  
4039 accepting Print-Job, Print-URI, and Create-Job requests. If the value is 'true', the printer is accepting jobs.  
4040 If the value is 'false', the Printer object is currently rejecting any jobs submitted to it. In this case, the  
4041 Printer object returns the 'server-error-not-accepting-jobs' status code.

4042 This value is independent of the "printer-state" and "printer-state-reasons" attributes because its value does  
4043 not affect the current job; rather it affects future jobs. This attribute, when 'false', causes the Printer to  
4044 reject jobs even when the "printer-state" is 'idle' or, when 'true', causes the Printer object to accept jobs  
4045 even when the "printer-state" is 'stopped'.

## 4046 4.4.24 queued-job-count (integer(0:MAX))

4047 This REQUIRED Printer attribute contains a count of the number of jobs that are either 'pending',  
4048 'processing', 'pending-held', or 'processing-stopped' and is set by the Printer object.

## 4049 4.4.25 printer-message-from-operator (text(127))

4050 This Printer attribute provides a message from an operator, system administrator or "intelligent" process to  
4051 indicate to the end user information or status of the printer, such as why it is unavailable or when it is  
4052 expected to be available.

## 4053 4.4.26 color-supported (boolean)

4054 This Printer attribute identifies whether the device is capable of any type of color printing at all, including  
4055 highlight color. All document instructions having to do with color are embedded within the document PDL  
4056 (none are external IPP attributes in IPP/1.1).

4057 Note: end-users are able to determine the nature and details of the color support by querying the "printer-  
4058 more-info-manufacturer" Printer attribute.

## 4059 4.4.27 reference-uri-schemes-supported (1setOf uriScheme)

4060 This Printer attribute specifies which URI schemes are supported for use in the "document-uri" operation  
4061 attribute of the Print-URI or Send-URI operation. If a Printer object supports these optional operations, it  
4062 MUST support the "reference-uri-schemes-supported" Printer attribute with at least the following schemed  
4063 URI value:

4064 'ftp': The Printer object will use an FTP 'get' operation as defined in RFC 2228 [RFC2228] using FTP  
4065 URLs as defined by [RFC2396] and[RFC2316].  
4066

4067 The Printer object MAY OPTIONALLY support other URI schemes (see section 4.1.6).

## 4068 4.4.28 pdl-override-supported (type2 keyword)

4069 This REQUIRED Printer attribute expresses the ability for a particular Printer implementation to either  
4070 attempt to override document data instructions with IPP attributes or not.

4071 This attribute takes on the following values:

- 4072 - 'attempted': This value indicates that the Printer object attempts to make the IPP attribute values take  
4073 precedence over embedded instructions in the document data, however there is no guarantee.
- 4074 - 'not-attempted': This value indicates that the Printer object makes no attempt to make the IPP attribute  
4075 values take precedence over embedded instructions in the document data.  
4076

4077 Section 15 contains a full description of how this attribute interacts with and affects other IPP attributes,  
4078 especially the "ipp-attribute-fidelity" attribute.

## 4079 4.4.29 printer-up-time (integer(1:MAX))

4080 This REQUIRED Printer attribute indicates the amount of time (in seconds) that this Printer instance has  
4081 been up and running. The value is a monotonically increasing value starting from 1 when the Printer object  
4082 is started-up (initialized, booted, etc.). This value is used to populate the Event Time Job Description Job  
4083 attributes "time-at-creation", "time-at-processing", and "time-at-completed" (see section 4.3.14).

4084 If the Printer object goes down at some value 'n', and comes back up, the implementation MAY:

- 4085 1. Know how long it has been down, and resume at some value greater than 'n', or
- 4086 2. Restart from 1.

4087 In other words, if the device or devices that the Printer object is representing are restarted or power cycled,  
4088 the Printer object MAY continue counting this value or MAY reset this value to 1 depending on  
4089 implementation. However, if the Printer object software ceases running, and restarts without knowing the  
4090 last value for "printer-up-time", the implementation MUST reset this value to 1. If this value is reset and  
4091 the Printer has persistent jobs, the Printer MUST reset the "time-at-xxx(integer) Event Time Job  
4092 Description attributes according to Section 4.3.14. An implementation MAY use both implementation  
4093 alternatives, depending on warm versus cold start, respectively.

## 4094 4.4.30 printer-current-time (dateTime)

4095 This Printer attribute indicates the current date and time. This value is used to populate the Event Time Job  
4096 Description attributes: "time-at-creation", "time-at-processing", and "time-at-completed" (see Section  
4097 4.3.14).

4098 The date and time is obtained on a "best efforts basis" and does not have to be that precise in order to work  
4099 in practice. A Printer implementation sets the value of this attribute by obtaining the date and time via  
4100 some implementation-dependent means, such as getting the value from a network time server, initialization  
4101 at time of manufacture, or setting by an administrator. See [IPP-IIG] for examples. If an implementation  
4102 supports this attribute and the implementation knows that it has not yet been set, then the implementation  
4103 MUST return the value of this attribute using the out-of-band 'no-value' meaning not configured. See the  
4104 beginning of section 4.1.

4105 The time zone of this attribute NEED NOT be the time zone used by people located near the Printer object  
4106 or device. The client MUST NOT expect that the time zone of any received 'dateTime' value to be in the  
4107 time zone of the client or in the time zone of the people located near the printer.

4108 The client SHOULD display any dateTime attributes to the user in client local time by converting the  
4109 'dateTime' value returned by the server to the time zone of the client, rather than using the time zone  
4110 returned by the Printer in attributes that use the 'dateTime' attribute syntax.

## 4111 4.4.31 multiple-operation-time-out (integer(1:MAX))

4112 This Printer attributes identifies the minimum time (in seconds) that the Printer object waits for additional  
4113 Send-Document or Send-URI operations to follow a still-open multi-document Job object before taking  
4114 any recovery actions, such as the ones indicated in section 3.3.1. If the Printer object supports the Create-  
4115 Job and Send-Document operations (see section 3.2.4 and 3.3.1), it MUST support this attribute.

4116 It is RECOMMENDED that vendors supply a value for this attribute that is between 60 and 240 seconds.  
4117 An implementation MAY allow a system administrator to set this attribute (by means outside this IPP/1.1  
4118 document). If so, the system administrator MAY be able to set values outside this range.

## 4119 4.4.32 compression-supported (1setOf type3 keyword)

4120 This REQUIRED Printer attribute identifies the set of supported compression algorithms for document  
4121 data. Compression only applies to the document data; compression does not apply to the encoding of the  
4122 IPP operation itself. The supported values are used to validate the client supplied "compression" operation  
4123 attributes in Print-Job, Send-Document, and Send-URI requests.

4124 Standard values are :

4125 'none': no compression is used.

4126 'deflate': ZIP public domain inflate/deflate) compression technology

4127 'gzip' GNU zip compression technology described in RFC 1952 [RFC1952].

4128 'compress': UNIX compression technology

4129



## 4130 4.4.33 job-k-octets-supported (rangeOfInteger(0:MAX))

4131 This Printer attribute specifies the upper and lower bounds of total sizes of jobs in K octets, i.e., in units of  
4132 1024 octets. The supported values are used to validate the client supplied "job-k-octets" operation attributes  
4133 in create requests. The corresponding job description attribute "job-k-octets" is defined in section 4.3.17.1.

## 4134 4.4.34 job-impressions-supported (rangeOfInteger(0:MAX))

4135 This Printer attribute specifies the upper and lower bounds for the number of impressions per job. The  
4136 supported values are used to validate the client supplied "job-impressions" operation attributes in create  
4137 requests. The corresponding job description attribute "job-impressions" is defined in section 4.3.17.2.

## 4138 4.4.35 job-media-sheets-supported (rangeOfInteger(0:MAX))

4139 This Printer attribute specifies the upper and lower bounds for the number of media sheets per job. The  
4140 supported values are used to validate the client supplied "job-media-sheets" operation attributes in create  
4141 requests. The corresponding Job attribute "job-media-sheets" is defined in section 4.3.17.3.

## 4142 4.4.36 pages-per-minute (integer(0:MAX))

4143 This Printer attributes specifies the nominal number of pages per minute to the nearest whole number which  
4144 may be generated by this printer (e.g., simplex, black-and-white). This attribute is informative, not a  
4145 service guarantee. Generally, it is the value used in the marketing literature to describe the device.

4146 A value of 0 indicates a device that takes more than two minutes to process a page.

## 4147 4.4.37 pages-per-minute-color (integer(0:MAX))

4148 This Printer attributes specifies the nominal number of pages per minute to the nearest whole number which  
4149 may be generated by this printer when printing color (e.g., simplex, color). For purposes of this attribute,  
4150 "color" means the same as for the "color-supported" attribute, namely, the device is capable of any type of  
4151 color printing at all, including highlight color. This attribute is informative, not a service guarantee.  
4152 Generally, it is the value used in the marketing literature to describe the color capabilities of this device.

4153 A value of 0 indicates a device that takes more than two minutes to process a page.

4154 If a color device has several color modes, it MAY use the pages-per-minute value for this attribute that  
4155 corresponds to the mode that produces the highest number.

4156 Black and white only printers MUST NOT support this attribute. If this attribute is present, then the "color-  
4157 supported" Printer description attribute MUST be present and have a 'true' value.

4158 The values of these two attributes returned by the Get-Printer-Attributes operation MAY be affected by the  
4159 "document-format" attribute supplied by the client in the Get-Printer-Attributes request. In other words, the  
4160 implementation MAY have different speeds depending on the document format being processed. See  
4161 section 3.2.5.1 Get-Printer-Attributes.

## 4162 5. Conformance

4163 This section describes conformance issues and requirements. This document introduces model entities such  
4164 as objects, operations, attributes, attribute syntaxes, and attribute values. These conformance sections  
4165 describe the conformance requirements which apply to these model entities.

## 4166 5.1 Client Conformance Requirements

4167 This section describes the conformance requirements for a client (see section 2.1), whether it be:

- 4168 1. contained within software controlled by an end user, e.g. activated by the "Print" menu item in an  
4169 application that sends IPP requests or
- 4170 2. the print server component that sends IPP requests to either an output device or another  
4171 "downstream" print server.

4172 A conforming client **MUST** support all **REQUIRED** operations as defined in this document. For each  
4173 attribute included in an operation request, a conforming client **MUST** supply a value whose type and value  
4174 syntax conforms to the requirements of the Model document as specified in Sections 3 and 4. A  
4175 conforming client **MAY** supply any registered extensions and/or private extensions in an operation request,  
4176 as long as they meet the requirements in Section 6.

4177 Otherwise, there are no conformance requirements placed on the user interfaces provided by IPP clients or  
4178 their applications. For example, one application might not allow an end user to submit multiple documents  
4179 per job, while another does. One application might first query a Printer object in order to supply a graphical  
4180 user interface (GUI) dialogue box with supported and default values whereas a different implementation  
4181 might not.

4182 When sending a request, an IPP client **NEED NOT** supply any attributes that are indicated as  
4183 **OPTIONALLY** supplied by the client.

4184 A client **MUST** be able to accept any of the attribute syntaxes defined in Section 4.1, including their full  
4185 range, that may be returned to it in a response from a Printer object. In particular for each attribute that the  
4186 client supports whose attribute syntax is 'text', the client **MUST** accept and process both the  
4187 'textWithoutLanguage' and 'textWithLanguage' forms. Similarly, for each attribute that the client supports  
4188 whose attribute syntax is 'name', the client **MUST** accept and process both the 'nameWithoutLanguage' and  
4189 'nameWithLanguage' forms. For presentation purposes, truncation of long attribute values is not  
4190 recommended. A recommended approach would be for the client implementation to allow the user to scroll  
4191 through long attribute values.

4192 A response **MAY** contain attribute groups, attributes, attribute syntaxes, values, and status codes that the  
4193 client does not expect. Therefore, a client implementation **MUST** gracefully handle such responses and not  
4194 refuse to inter-operate with a conforming Printer that is returning registered or private extensions, including  
4195 attribute groups, attributes, attribute syntaxes, attribute values, and status codes that conform to Section 6.  
4196 Clients may choose to ignore any parameters, attributes, attribute syntaxes, or values that they do not  
4197 understand.

4198 While a client is sending data to a printer, it SHOULD do its best to prevent a channel from being closed by  
 4199 a lower layer when the channel is blocked (i.e. flow-controlled off) for whatever reason, e.g. 'out of paper'  
 4200 or 'job ahead hasn't freed up enough memory'. However, the layer that launched the print submission (e.g.  
 4201 an end user) MAY close the channel in order to cancel the job. When a client closes a channel, a Printer  
 4202 MAY print all or part of the received portion of the document. See the "Encoding and Transport" document  
 4203 [IPP-PRO] for more details.

4204 A client MUST support Client Authentication as defined in the IPP/1.1 Encoding and Transport document  
 4205 [IPP-PRO]. A client SHOULD support Operation Privacy and Server Authentication as defined in the  
 4206 IPP/1.1 Encoding and Transport document [IPP-PRO]. See also section 8 of this document.

## 4207 5.2 IPP Object Conformance Requirements

4208 This section specifies the conformance requirements for conforming implementations of IPP objects (see  
 4209 section 2). These requirements apply to an IPP object whether it is:

- 4210 (1) an (embedded) device component that accepts IPP requests and controls the device or
- 4211 (2) a component of a print server that accepts IPP requests (where the print server control one or  
 4212 more networked devices using IPP or other protocols).

### 4213 5.2.1 Objects

4214 Conforming implementations MUST implement all of the model objects as defined in this document in the  
 4215 indicated sections:

- 4216 Section 2.1 - Printer Object
- 4217 Section 2.2 - Job Object

### 4218 5.2.2 Operations

4219 Conforming IPP object implementations MUST implement all of the REQUIRED model operations,  
 4220 including REQUIRED responses, as defined in this document in the indicated sections:

4221 For a Printer object:

4222	Print-Job (section 3.2.1)	REQUIRED
4223	Print-URI (section 3.2.2)	OPTIONAL
4224	Validate-Job (section 3.2.3)	REQUIRED
4225	Create-Job (section 3.2.4)	OPTIONAL
4226	Get-Printer-Attributes (section 3.2.5)	REQUIRED
4227	Get-Jobs (section 3.2.6)	REQUIRED
4228	Pause-Printer (section 3.2.7)	OPTIONAL
4229	Resume-Printer (section 3.2.8)	OPTIONAL
4230	Purge-Jobs (section 3.2.9)	OPTIONAL

4231

4232 For a Job object:

4233	Send-Document (section 3.3.1)	OPTIONAL
------	-------------------------------	----------

4234	Send-URI (section 3.3.2)	OPTIONAL
4235	Cancel-Job (section 3.3.3)	REQUIRED
4236	Get-Job-Attributes (section 3.3.4)	REQUIRED
4237	Hold-Job (section 3.3.5)	OPTIONAL
4238	Release-Job (section 3.3.6)	OPTIONAL
4239	Restart-Job (section 3.3.7)	OPTIONAL
4240		

4241 Conforming IPP objects MUST support all REQUIRED operation attributes and all values of such  
4242 attributes if so indicated in the description. Conforming IPP objects MUST ignore all unsupported or  
4243 unknown operation attributes or operation attribute groups received in a request, but MUST reject a request  
4244 that contains a supported operation attribute that contains an unsupported value.

4245 Conforming IPP objects MAY return operation responses that contain attributes groups, attributes names,  
4246 attribute syntaxes, attribute values, and status codes that are extensions to this standard. The additional  
4247 attribute groups MAY occur in any order.

4248 The following section on object attributes specifies the support required for object attributes.

### 4249 5.2.3 IPP Object Attributes

4250 Conforming IPP objects MUST support all of the REQUIRED object attributes, as defined in this document  
4251 in the indicated sections.

4252 If an object supports an attribute, it MUST support only those values specified in this document or through  
4253 the extension mechanism described in section 5.2.4. It MAY support any non-empty subset of these values.  
4254 That is, it MUST support at least one of the specified values and at most all of them.

### 4255 5.2.4 Versions

4256 Clients MUST meet the conformance requirements for clients specified in this document and [IPP-PRO]  
4257 and SHOULD also support version 1.0, i.e., SHOULD meet the conformance requirements for clients as  
4258 specified in [RFC2566] and [RFC2565].

4259 IPP Printer and Job objects MUST meet the conformance requirements for IPP objects specified in this  
4260 document and [IPP-PRO]. For interoperability with IPP/1.0 clients, IPP/1.1 objects SHOULD also meet  
4261 the conformance requirements for IPP objects as specified in [RFC2566] and [RFC2565].

4262 Clients MUST send requests containing a "version-number" parameter with a '1.1' value and SHOULD try  
4263 supplying alternate version numbers if they receive a 'server-error-version-not-supported' error return in a  
4264 response.

4265 IPP objects MUST accept requests containing a "version-number" parameter with a '1.1' value (or reject the  
4266 request if the operation is not supported). IPP objects SHOULD accept any request with the major version  
4267 '1' (or reject the request if the operation is not supported). See section 3.1.8.

## 4268 5.2.5 Extensions

4269 A conforming IPP object MAY support registered extensions and private extensions, as long as they meet  
4270 the requirements specified in Section 6.

4271 For each attribute included in an operation response, a conforming IPP object MUST return a value whose  
4272 type and value syntax conforms to the requirement of the Model document as specified in Sections 3 and 4.

## 4273 5.2.6 Attribute Syntaxes

4274 An IPP object MUST be able to accept any of the attribute syntaxes defined in Section 4.1, including their  
4275 full range, in any operation in which a client may supply attributes or the system administrator may  
4276 configure attributes (by means outside the scope of this IPP/1.1 document). In particular for each attribute  
4277 that the IPP object supports whose attribute syntax is 'text', the IPP object MUST accept and process both  
4278 the 'textWithoutLanguage' and 'textWithLanguage' forms. Similarly, for each attribute that the IPP object  
4279 supports whose attribute syntax is 'name', the IPP object MUST accept and process both the  
4280 'nameWithoutLanguage' and 'nameWithLanguage' forms. Furthermore, an IPP object MUST return  
4281 attributes to the client in operation responses that conform to the syntax specified in Section 4.1, including  
4282 their full range if supplied previously by a client.

## 4283 5.2.7 Security

4284 An IPP Printer implementation SHOULD contain support for Client Authentication as defined in the  
4285 IPP/1.1 Encoding and Transport document [IPP-PRO]. A Printer implementation MAY allow an  
4286 administrator to configure the Printer so that all, some, or none of the users are authenticated. See also  
4287 section 8 of this document.

4288 An IPP Printer implementation SHOULD contain support for Operation Privacy and Server Authentication  
4289 as defined in the IPP/1.1 Encoding and Transport document [IPP-PRO]. A Printer implementation MAY  
4290 allow an administrator to configure the degree of support for Operation Privacy and Server Authentication.  
4291 See also section 8 of this document.

4292 Security MUST NOT be compromised when a client supplies a lower "version-number" parameter in a  
4293 request. For example, if an IPP/1.1 conforming Printer object accepts version '1.0' requests and is  
4294 configured to enforce Digest Authentication, it MUST do the same for a version '1.0' request.

## 4295 5.3 Charset and Natural Language Requirements

4296 All clients and IPP objects MUST support the 'utf-8' charset as defined in section 4.1.7.

4297 IPP objects MUST be able to accept any client request which correctly uses the "attributes-natural-  
4298 language" operation attribute or the Natural Language Override mechanism on any individual attribute  
4299 whether or not the natural language is supported by the IPP object. If an IPP object supports a natural  
4300 language, then it MUST be able to translate (perhaps by table lookup) all generated 'text' or 'name' attribute  
4301 values into one of the supported languages (see section 3.1.4). That is, the IPP object that supports a

4302 natural language NEED NOT be a general purpose translator of any arbitrary 'text' or 'name' value supplied  
4303 by the client into that natural language. However, the object MUST be able to translate (automatically  
4304 generate) any of its own attribute values and messages into that natural language.

## 4305 6. IANA Considerations (registered and private extensions)

4306 This section describes how IPP can be extended to allow the following registered and private extensions to  
4307 IPP:

- 4308 1. keyword attribute values
- 4309 2. enum attribute values
- 4310 3. attributes
- 4311 4. attribute syntaxes
- 4312 5. operations
- 4313 6. attribute groups
- 4314 7. status codes

4315

4316 Extensions registered for use with IPP/1.1 are OPTIONAL for client and IPP object conformance to the  
4317 IPP/1.1 Model document.

4318 These extension procedures are aligned with the guidelines as set forth by the IESG [IANA-CON]. Section  
4319 11 describes how to propose new registrations for consideration. IANA will reject registration proposals  
4320 that leave out required information or do not follow the appropriate format described in Section 11. IPP/1.1  
4321 may also be extended by an appropriate RFC that specifies any of the above extensions.

### 4322 6.1 Typed 'keyword' and 'enum' Extensions

4323 IPP allows for 'keyword' and 'enum' extensions (see sections 4.1.2.3 and 4.1.4). This document uses  
4324 prefixes to the 'keyword' and 'enum' basic attribute syntax type in order to communicate extra information  
4325 to the reader through its name. This extra information is not represented in the protocol because it is  
4326 unimportant to a client or Printer object. The list below describes the prefixes and their meaning.

4327 "type1": This IPP specification document must be revised to add a new keyword or a new enum. No  
4328 private keywords or enums are allowed.

4329

4330 "type2": Implementers can, at any time, add new keyword or enum values by proposing the complete  
4331 specification to IANA:

4332

4333 iana@iana.org

4334

4335 IANA will forward the registration proposal to the IPP Designated Expert who will review the  
4336 proposal with a mailing list that the Designated Expert keeps for this purpose. Initially, that list will  
4337 be the mailing list used by the IPP WG:

4338

4339 ipp@pwg.org

4340

4341 even after the IPP WG is disbanded as permitted by [IANA-CON]. The IPP Designated Expert is  
4342 appointed by the IESG Area Director responsible for IPP, according to [IANA-CON].

4343

4344 When a type2 keyword or enum is approved, the IPP Designated Expert becomes the point of  
4345 contact for any future maintenance that might be required for that registration.

4346

4347 "type3": Implementers can, at any time, add new keyword and enum values by submitting the complete  
4348 specification to IANA as for type2 who will forward the proposal to the IPP Designated Expert.  
4349 While no additional technical review is required, the IPP Designated Expert may, at his/her  
4350 discretion, forward the proposal to the same mailing list as for type2 registrations for advice and  
4351 comment.

4352

4353 When a type3 keyword or enum is approved by the IPP Designated Expert, the original proposer  
4354 becomes the point of contact for any future maintenance that might be required for that registration.

4355

4356 For type2 and type3 keywords, the proposer includes the name of the keyword in the registration proposal  
4357 and the name is part of the technical review.

4358 After type2 and type3 enums specifications are approved, the IPP Designated Expert in consultation with  
4359 IANA assigns the next available enum number for each enum value.

4360 IANA will publish approved type2 and type3 keyword and enum attributes value registration specifications  
4361 in:

4362 ftp.isi.edu/iana/assignments/ipp/attribute-values/xxx/yyy.txt

4363 where xxx is the attribute name that specifies the initial values and yyy.txt is a descriptive file name that  
4364 contains one or more enums or keywords approved at the same time. For example, if several additional  
4365 enums for stapling are approved for use with the "finishings" attribute (and "finishings-default" and  
4366 "finishings-supported" attributes), IANA will publish the additional values in the file:

4367 ftp.isi.edu/iana/assignments/ipp/attribute-values/finishings/stapling.txt

4368 Note: Some attributes are defined to be: 'type3 keywords' | 'name' which allows for attribute values to be  
4369 extended by a site administrator with administrator defined names. Such names are not registered with  
4370 IANA.

4371 By definition, each of the three types above assert some sort of registry or review process in order for  
4372 extensions to be considered valid. Each higher numbered level (1, 2, 3) tends to be decreasingly less  
4373 stringent than the previous level. Therefore, any typeN value MAY be registered using a process for some  
4374 typeM where M is less than N, however such registration is NOT REQUIRED. For example, a type3 value  
4375 MAY be registered in a type 1 manner (by being included in a future version of an IPP specification),  
4376 however, it is NOT REQUIRED.

4377 This document defines keyword and enum values for all of the above types, including type3 keywords.

4378 For private (unregistered) keyword extensions, implementers SHOULD use keywords with a suitable  
4379 distinguishing prefix, such as "xxx-" where xxx is the (lowercase) fully qualified company name registered  
4380 with IANA for use in domain names [RFC1035]. For example, if the company XYZ Corp. had obtained  
4381 the domain name "XYZ.com", then a private keyword 'abc' would be: 'xyz.com-abc'.

4382 Note: RFC 1035 [RFC1035] indicates that while upper and lower case letters are allowed in domain names,  
4383 no significance is attached to the case. That is, two names with the same spelling but different case are to  
4384 be treated as if identical. Also, the labels in a domain name must follow the rules for ARPANET host  
4385 names: They must start with a letter, end with a letter or digit, and have as interior characters only letters,  
4386 digits, and hyphen. Labels must be 63 characters or less. Labels are separated by the "." character.

4387 For private (unregistered) enum extension, implementers MUST use values in the reserved integer range  
4388 which is  $2^{30}$  to  $2^{31}-1$ .

## 4389 6.2 Attribute Extensibility

4390 Attribute names are type2 keywords. Therefore, new attributes may be registered and have the same status  
4391 as attributes in this document by following the type2 extension rules. For private (unregistered) attribute  
4392 extensions, implementers SHOULD use keywords with a suitable distinguishing prefix as described in  
4393 Section 6.1.

4394 IANA will publish approved attribute registration specifications as separate files:

4395 `ftp.isi.edu/iana/assignments/ipp/attributes/xxx-yyy.txt`

4396 where "xxx-yyy" is the new attribute name.

4397 If a new Printer object attribute is defined and its values can be affected by a specific document format, its  
4398 specification needs to contain the following sentence:

4399 "The value of this attribute returned in a Get-Printer-Attributes response MAY depend on the  
4400 "document-format" attribute supplied (see Section 3.2.5.1)."

4401 If the specification does not, then its value in the Get-Printer-Attributes response MUST NOT depend on  
4402 the "document-format" supplied in the request. When a new Job Template attribute is registered, the value  
4403 of the Printer attributes MAY vary with "document-format" supplied in the request without the  
4404 specification having to indicate so.



## 4405 6.3 Attribute Syntax Extensibility

4406 Attribute syntaxes are like type2 enums. Therefore, new attribute syntaxes may be registered and have the  
4407 same status as attribute syntaxes in this document by following the type2 extension rules described in  
4408 Section 6.1. The value codes that identify each of the attribute syntaxes are assigned in the "Encoding and  
4409 Transport" document [IPP-PRO], including a designated range for private, experimental use.

4410 For attribute syntaxes, the IPP Designated Expert in consultation with IANA assigns the next attribute  
4411 syntax code in the appropriate range as specified in [IPP-PRO]. IANA will publish approved attribute  
4412 syntax registration specifications as separate files:

4413 `ftp.isi.edu/iana/assignments/ipp/attribute-syntaxes/xxx-yyy.txt`

4414 where 'xxx-yyy' is the new attribute syntax name.

## 4415 6.4 Operation Extensibility

4416 Operations may also be registered following the type2 procedures described in Section 6.1, though major  
4417 new operations will usually be done by a new standards track RFC that augments this document. For  
4418 private (unregistered) operation extensions, implementers **MUST** use the range for the "operation-id" in  
4419 requests specified in Section 4.4.15 "operations-supported" Printer attribute.

4420 For operations, the IPP Designated Expert in consultation with IANA assigns the next operation-id code as  
4421 specified in Section 4.4.15. IANA will publish approved operation registration specifications as separate  
4422 files:

4423 `ftp.isi.edu/iana/assignments/ipp/operations/Xxx-Yyy.txt`

4424 where "Xxx-Yyy" is the new operation name.

## 4425 6.5 Attribute Groups

4426 Attribute groups passed in requests and responses may be registered following the type2 procedures  
4427 described in Section 6.1. The tags that identify each of the attribute groups are assigned in [IPP-PRO].

4428 For attribute groups, the IPP Designated Expert in consultation with IANA assigns the next attribute group  
4429 tag code in the appropriate range as specified in [IPP-PRO]. IANA will publish approved attribute group  
4430 registration specifications as separate files:

4431 `ftp.isi.edu/iana/assignments/ipp/attribute-group-tags/xxx-yyy-tag.txt`

4432 where 'xxx-yyy-tag' is the new attribute group tag name.

## 4433 6.6 Status Code Extensibility

4434 Operation status codes may also be registered following the type2 procedures described in Section 6.1. The  
4435 values for status codes are allocated in ranges as specified in Section 14 for each status code class:

- 4436 "informational" - Request received, continuing process
- 4437 "successful" - The action was successfully received, understood, and accepted
- 4438 "redirection" - Further action must be taken in order to complete the request
- 4439 "client-error" - The request contains bad syntax or cannot be fulfilled
- 4440 "server-error" - The IPP object failed to fulfill an apparently valid request

4441

4442 For private (unregistered) operation status code extensions, implementers **MUST** use the top of each range  
4443 as specified in Section 13.

4444 For operation status codes, the IPP Designated Expert in consultation with IANA assigns the next status  
4445 code in the appropriate class range as specified in Section 13. IANA will publish approved status code  
4446 registration specifications as separate files:

4447 ftp.isi.edu/iana/assignments/ipp/status-codes/xxx-yyy.txt

4448 where "xxx-yyy" is the new operation status code keyword.

## 4449 6.7 Registration of MIME types/sub-types for document-formats

4450 The "document-format" attribute's syntax is 'mimeMediaType'. This means that valid values are Internet  
4451 Media Types (see Section 4.1.9). RFC 2045 [RFC2045] defines the syntax for valid Internet media types.  
4452 IANA is the registry for all Internet media types.

## 4453 6.8 Registration of charsets for use in 'charset' attribute values

4454 The "attributes-charset" attribute's syntax is 'charset'. This means that valid values are charsets names.  
4455 When a charset in the IANA registry has more than one name (alias), the name labeled as "(preferred  
4456 MIME name)", if present, **MUST** be used (see Section 4.1.7). IANA is the registry for charsets following  
4457 the procedures of [RFC2278].

## 4458 7. Internationalization Considerations

4459 Some of the attributes have values that are text strings and names which are intended for human  
4460 understanding rather than machine understanding (see the 'text' and 'name' attribute syntaxes in Sections  
4461 4.1.1 and 4.1.2).

4462 In each operation request, the client

- 4463 - identifies the charset and natural language of the request which affects each supplied 'text' and 'name'  
4464 attribute value, and

4465 - requests the charset and natural language for attributes returned by the IPP object in operation  
4466 responses (as described in Section 3.1.4.1).  
4467

4468 In addition, the client MAY separately and individually identify the Natural Language Override of a  
4469 supplied 'text' or 'name' attribute using the 'textWithLanguage' and 'nameWithLanguage' technique  
4470 described section 4.1.1.2 and 4.1.2.2 respectively.

4471 All IPP objects MUST support the UTF-8 [RFC2279] charset in all 'text' and 'name' attributes supported. If  
4472 an IPP object supports more than the UTF-8 charset, the object MUST convert between them in order to  
4473 return the requested charset to the client according to Section 3.1.4.2. If an IPP object supports more than  
4474 one natural language, the object SHOULD return 'text' and 'name' values in the natural language requested  
4475 where those values are generated by the Printer (see Section 3.1.4.1).

4476 For Printers that support multiple charsets and/or multiple natural languages in 'text' and 'name' attributes,  
4477 different jobs may have been submitted in differing charsets and/or natural languages. All responses MUST  
4478 be returned in the charset requested by the client. However, the Get-Jobs operation uses the  
4479 'textWithLanguage' and 'nameWithLanguage' mechanism to identify the differing natural languages with  
4480 each job attribute returned.

4481 The Printer object also has configured charset and natural language attributes. The client can query the  
4482 Printer object to determine the list of charsets and natural languages supported by the Printer object and  
4483 what the Printer object's configured values are. See the "charset-configured", "charset-supported", "natural-  
4484 language-configured", and "generated-natural-language-supported" Printer description attributes for more  
4485 details.

4486 The "charset-supported" attributed identifies the supported charsets. If a charset is supported, the IPP  
4487 object MUST be capable of converting to and from that charset into any other supported charset. In many  
4488 cases, an IPP object will support only one charset and it MUST be the UTF-8 charset.

4489 The "charset-configured" attribute identifies the one supported charset which is the native charset given the  
4490 current configuration of the IPP object (administrator defined).

4491 The "generated-natural-language-supported" attribute identifies the set of supported natural languages for  
4492 generated messages; it is not related to the set of natural languages that must be accepted for client supplied  
4493 'text' and 'name' attributes. For client supplied 'text' and 'name' attributes, an IPP object MUST accept ALL  
4494 supplied natural languages. Just because a Printer object is currently configured to support 'en-us' natural  
4495 language does not mean that the Printer object should reject a job if the client supplies a job name that is in  
4496 'fr-ca'.

4497 The "natural-language-configured" attribute identifies the one supported natural language for generated  
4498 messages which is the native natural language given the current configuration of the IPP object  
4499 (administrator defined).

4500 Attributes of type 'text' and 'name' are populated from different sources. These attributes can be categorized  
4501 into following groups (depending on the source of the attribute):

- 4502 1. Some attributes are supplied by the client (e.g., the client supplied "job-name", "document-name",  
4503 and "requesting-user-name" operation attributes along with the corresponding Job object's "job-  
4504 name" and "job-originating-user-name" attributes). The IPP object MUST accept these attributes in  
4505 any natural language no matter what the set of supported languages for generated messages  
4506 2. Some attributes are supplied by the system administrator (e.g., the Printer object's "printer-name" and  
4507 "printer-location" attributes). These too can be in any natural language. If the natural language for  
4508 these attributes is different than what a client requests, then they must be reported using the Natural  
4509 Language Override mechanism.  
4510 3. Some attributes are supplied by the device manufacturer (e.g., the Printer object's "printer-make-and-  
4511 model" attribute). These too can be in any natural language. If the natural language for these  
4512 attributes is different than what a client requests, then they must be reported using the Natural  
4513 Language Override mechanism.  
4514 4. Some attributes are supplied by the operator (e.g., the Job object's "job-message-from-operator"  
4515 attribute). These too can be in any natural language. If the natural language for these attributes is  
4516 different than what a client requests, then they must be reported using the Natural Language  
4517 Override mechanism.  
4518 5. Some attributes are generated by the IPP object (e.g., the Job object's "job-state-message" attribute,  
4519 the Printer object's "printer-state-message" attribute, and the "status-message" operation attribute).  
4520 These attributes can only be in one of the "generated-natural-language-supported" natural  
4521 languages. If a client requests some natural language for these attributes other than one of the  
4522 supported values, the IPP object SHOULD respond using the value of the "natural-language-  
4523 configured" attribute (using the Natural Language Override mechanism if needed).  
4524

4525 The 'text' and 'name' attributes specified in this version of this document (additional ones will be registered  
4526 according to the procedures in Section 6) are:

Attributes	Source
Operation Attributes:	
job-name (name)	client
document-name (name)	client
requesting-user-name (name)	client
status-message	Job or Printer object
Job Template Attributes:	
job-hold-until (keyword   name)	client matches administrator-configured
job-hold-until-default (keyword   name)	client matches administrator-configured
job-hold-until-supported (keyword   name)	client matches administrator-configured
job-sheets (keyword   name)	client matches administrator-configured
job-sheets-default (keyword   name)	client matches administrator-configured
job-sheets-supported (keyword   name)	client matches administrator-configured
media (keyword   name)	client matches administrator-configured
media-default (keyword   name)	client matches administrator-configured
media-supported (keyword   name)	client matches administrator-configured
media-ready (keyword   name)	client matches administrator-configured
Job Description Attributes:	
job-name (name)	client or Printer object
job-originating-user-name (name)	Printer object
job-state-message (text)	Job or Printer object
output-device-assigned (name(127))	administrator
job-message-from-operator (text(127))	operator
Printer Description Attributes:	
printer-name (name(127))	administrator
printer-location (text(127))	administrator
printer-info (text(127))	administrator
printer-make-and-model (text(127))	administrator or manufacturer
printer-state-message (text)	Printer object
printer-message-from-operator (text(127))	operator

## 4527 8. Security Considerations

4528 It is difficult to anticipate the security risks that might exist in any given IPP environment. For example, if  
 4529 IPP is used within a given corporation over a private network, the risks of exposing document data may be  
 4530 low enough that the corporation will choose not to use encryption on that data. However, if the connection

4531 between the client and the IPP object is over a public network, the client may wish to protect the content of  
4532 the information during transmission through the network with encryption.

4533 Furthermore, the value of the information being printed may vary from one IPP environment to the next.  
4534 Printing payroll checks, for example, would have a different value than printing public information from a  
4535 file. There is also the possibility of denial-of-service attacks, but denial-of-service attacks against printing  
4536 resources are not well understood and there is no published precedents regarding this scenario.

4537 Once the authenticated identity of the requester has been supplied to the IPP object, the object uses that  
4538 identity to enforce any authorization policy that might be in place. For example, one site's policy might be  
4539 that only the job owner is allowed to cancel a job. The details and mechanisms to set up a particular access  
4540 control policy are not part of IPP/1.1, and must be established via some other type of administrative or  
4541 access control framework. However, there are operation status codes that allow an IPP server to return  
4542 information back to a client about any potential access control violations for an IPP object.

4543 During a create operation, the client's identity is recorded in the Job object in an implementation-defined  
4544 attribute. This information can be used to verify a client's identity for subsequent operations on that Job  
4545 object in order to enforce any access control policy that might be in effect. See section 8.3 below for more  
4546 details.

4547 Since the security levels or the specific threats that any given IPP system administrator may be concerned  
4548 with cannot be anticipated, IPP **MUST** be capable of operating with different security mechanisms and  
4549 security policies as required by the individual installation. Security policies might vary from very strong, to  
4550 very weak, to none at all, and corresponding security mechanisms will be required.

## 4551 8.1 Security Scenarios

4552 The following sections describe specific security attacks for IPP environments. Where examples are  
4553 provided they should be considered illustrative of the environment and not an exhaustive set. Not all of  
4554 these environments will necessarily be addressed in initial implementations of IPP.

### 4555 8.1.1 Client and Server in the Same Security Domain

4556 This environment is typical of internal networks where traditional office workers print the output of  
4557 personal productivity applications on shared work-group printers, or where batch applications print their  
4558 output on large production printers. Although the identity of the user may be trusted in this environment, a  
4559 user might want to protect the content of a document against such attacks as eavesdropping, replaying or  
4560 tampering.

### 4561 8.1.2 Client and Server in Different Security Domains

4562 Examples of this environment include printing a document created by the client on a publicly available  
4563 printer, such as at a commercial print shop; or printing a document remotely on a business associate's  
4564 printer. This latter operation is functionally equivalent to sending the document to the business associate as  
4565 a facsimile. Printing sensitive information on a Printer in a different security domain requires strong

4566 security measures. In this environment authentication of the printer is required as well as protection against  
4567 unauthorized use of print resources. Since the document crosses security domains, protection against  
4568 eavesdropping and document tampering are also required. It will also be important in this environment to  
4569 protect Printers against "spamming" and malicious document content.

### 4570 8.1.3 Print by Reference

4571 When the document is not stored on the client, printing can be done by reference. That is, the print request  
4572 can contain a reference, or pointer, to the document instead of the actual document itself (see sections 3.2.2  
4573 and 3.3.2). Standard methods currently do not exist for remote entities to "assume" the credentials of a  
4574 client for forwarding requests to a 3rd party. It is anticipated that Print-By-Reference will be used to access  
4575 "public" documents and that sophisticated methods for authenticating "proxies" is not specified in this  
4576 document.

### 4577 8.2 URIs in Operation, Job, and Printer attributes

4578 The "printer-uri-supported" attribute contains the Printer object's URI(s). Its companion attribute, "uri-  
4579 security-supported", identifies the security mechanism used for each URI listed in the "printer-uri-  
4580 supported" attribute. For each Printer operation request, a client MUST supply only one URI in the  
4581 "printer-uri" operation attribute. In other words, even though the Printer supports more than one URI, the  
4582 client only interacts with the Printer object using one of its URIs. This duality is not needed for Job objects,  
4583 since the Printer object is the factory for Job objects, and the Printer object will generate the correct URI  
4584 for new Job objects depending on the Printer object's security configuration.

### 4585 8.3 URIs for each authentication mechanisms

4586 Each URI has an authentication mechanism associated with it. If the URI is the *i*'th element of "printer-uri-  
4587 supported", then authentication mechanism is the *i* th element of "uri-authentication-supported". For a list  
4588 of possible authentication mechanisms, see section 4.4.2.

4589 The Printer object uses an authentication mechanism to determine the name of the user performing an  
4590 operation. This user is called the "authenticated user". The credibility of authentication depends on the  
4591 mechanism that the Printer uses to obtain the user's name. When the authentication mechanism is 'none', all  
4592 authenticated users are "anonymous".

4593 During job creation operations, the Printer initializes the value of the "job-originating-user-name" attribute  
4594 (see section 4.3.6) to be the authenticated user. The authenticated user in this case is called the "job-owner".

4595 If an implementation can be configured to support more than one authentication mechanism, then it MUST  
4596 implement rules for determining equality of authenticated user names which have been authenticated via  
4597 different authentication mechanisms. One possible policy is that identical names that are authenticated via  
4598 different mechanism are different. For example, a user can cancel his job only if he uses the same  
4599 authentication mechanism for both Cancel-Job and Print-Job. Another policy is that identical names that  
4600 are authenticated via different mechanism are the same if the authentication mechanism for the later  
4601 operation is not less strong than the authentication mechanism for the earlier job creation operation. For

4602 example, a user can cancel his job only if he uses the same or stronger authentication mechanism for  
4603 Cancel-Job and Print-Job. With this second policy a job submitted via 'requesting-user-name' authentication  
4604 could be cancelled via 'digest' authentication. With the first policy, the job could not be cancelled in this  
4605 way.

4606 A client is able to determine the authentication mechanism used to create a job. It is the *i*'th value of the  
4607 Printer's "uri-authentication-supported" attribute (see section 4.4.2), where *i* is the index of the element of  
4608 the Printer's "printer-uri-supported" attribute (see section 4.4.1) equal to the job's "job-printer-uri" attribute  
4609 (see section 4.3.3).

#### 4610 8.4 Restricted Queries

4611 In many IPP operations, a client supplies a list of attributes to be returned in the response. For security  
4612 reasons, an IPP object may be configured not to return all attributes (or all values) that a client requests.  
4613 The job attributes returned MAY depend on whether the requesting user is the same as the user that  
4614 submitted the job. The IPP object MAY even return none of the requested attributes. In such cases, the  
4615 status returned is the same as if the object had returned all requested attributes. The client cannot tell by  
4616 such a response whether the requested attribute was present or absent on the object.

#### 4617 8.5 Operations performed by operators and system administrators

4618 For the three printer operations Pause-Printer, Resume-Printer, and Purge-Jobs (see sections 3.2.7, 3.2.8 and  
4619 3.2.9), the requesting user is intended to be an operator or administrator of the Printer object (see section 1).  
4620 For operations on jobs, the requesting user is intended to be the job owner or may be an operator or  
4621 administrator of the Printer object. The means for authorizing an operator or administrator of the Printer  
4622 object are not specified in this document.

#### 4623 8.6 Queries on jobs submitted using non-IPP protocols

4624 If the device that an IPP Printer is representing is able to accept jobs using other job submission protocols  
4625 in addition to IPP, it is RECOMMENDED that such an implementation at least allow such "foreign" jobs to  
4626 be queried using Get-Jobs returning "job-id" and "job-uri" as 'unknown'. Such an implementation NEED  
4627 NOT support all of the same IPP job attributes as for IPP jobs. The IPP object returns the 'unknown' out-of-  
4628 band value for any requested attribute of a foreign job that is supported for IPP jobs, but not for foreign  
4629 jobs.

4630 It is further RECOMMENDED, that the IPP Printer generate "job-id" and "job-uri" values for such "foreign  
4631 jobs", if possible, so that they may be targets of other IPP operations, such as Get-Job-Attributes and  
4632 Cancel-Job. Such an implementation also needs to deal with the problem of authentication of such foreign  
4633 jobs. One approach would be to treat all such foreign jobs as belonging to users other than the user of the  
4634 IPP client. Another approach would be for the foreign job to belong to 'anonymous'. Only if the IPP client  
4635 has been authenticated as an operator or administrator of the IPP Printer object, could the foreign jobs be  
4636 queried by an IPP request. Alternatively, if the security policy is to allow users to query other users' jobs,  
4637 then the foreign jobs would also be visible to an end-user IPP client using Get-Jobs and Get-Job-Attributes.



## 4638 9. References

## 4639 [ASCII]

4640 Coded Character Set - 7-bit American Standard Code for Information Interchange (ASCII), ANSI  
4641 X3.4-1986. This standard is the specification of the US-ASCII charset.

## 4642 [BCP-11]

4643 [Bradner S.](#), [Hovey R.](#), "The Organizations Involved in the IETF Standards Process", 1996/10/29  
4644 (RFC 2028)

## 4645 [HTPP]

4646 J. Barnett, K. Carter, R. DeBry, "Initial Draft - Hypertext Printing Protocol - HTPP/1.0", October  
4647 1996, <ftp://ftp.pwg.org/pub/pwg/ipp/historic/http/overview.ps.gz>

## 4648 [IANA-CON]

4649 Narte, T. and Alvestrand, H.T.: Guidelines for Writing an IANA Considerations Section in RFCs,  
4650 Work in Progress, draft-iesg-iana-considerations-04.txt, May 21, 1998.

## 4651 [IANA-CS]

4652 IANA Registry of Coded Character Sets: <ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets>

## 4653 [IANA-MT]

4654 IANA Registry of Media Types: <ftp://ftp.isi.edu/in-notes/iana/assignments/media-types/>

## 4655 [IPP-IIG]

4656 Hastings, T., Manros, C., "Internet Printing Protocol/1.1: draft-ietf-ipp-implementers-guide-v11-  
4657 ???.txt, ?? 1999, work in progress.

## 4658 [IPP-IIG1.0]

4659 Hastings, T., Manros, C., "Internet Printing Protocol/1.0: Implementer's Guide", draft-ietf-ipp-  
4660 implementers-guide-01.txt, February 1999, work in progress.

## 4661 [IPP-PRO]

4662 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.1: Encoding and  
4663 Transport", draft-ietf-ipp-protocol-v11-01.txt, May, 1999.

## 4664 [ISO10646-1]

4665 ISO/IEC 10646-1:1993, "Information technology -- Universal Multiple-Octet Coded Character Set  
4666 (UCS) - Part 1: Architecture and Basic Multilingual Plane, JTC1/SC2."

## 4667 [ISO8859-1]

4668 ISO/IEC 8859-1:1987, "Information technology -- 8-bit One-Byte Coded Character Set - Part 1:  
4669 Latin Alphabet Nr 1", 1987, JTC1/SC2.

## 4670 [ISO10175]

4671 ISO/IEC 10175 Document Printing Application (DPA), June 1996.

- 4672 [LDPA]  
4673 T. Hastings, S. Isaacson, M. MacKay, C. Manros, D. Taylor, P. Zehler, "LDPA - Lightweight  
4674 Document Printing Application", October 1996,  
4675 <ftp://ftp.pwg.org/pub/pwg/ipp/historic/ldpa/ldpa8.pdf.gz>
- 4676 [P1387.4]  
4677 Kirk, M. (editor), POSIX System Administration - Part 4: Printing Interfaces, POSIX 1387.4 D8,  
4678 1994.
- 4679 [PSIS] Herriot, R. (editor), X/Open A Printing System Interoperability Specification (PSIS), August 1995.
- 4680 [PWG]  
4681 Printer Working Group, <http://www.pwg.org>.
- 4682 [RFC1035]  
4683 P. Mockapetris, "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION", RFC 1035,  
4684 November 1987.
- 4685 [RFC1179]  
4686 McLaughlin, L. III, (editor), "Line Printer Daemon Protocol" RFC 1179, August 1990.
- 4687 [RFC1759]  
4688 Smith, R., Wright, F., Hastings, T., Zilles, S., and Gyllenskog, J., "Printer MIB", RFC 1759, March  
4689 1995.
- 4690 [RFC1766]  
4691 H. Alvestrand, "Tags for the Identification of Languages", RFC 1766, March 1995.
- 4692 [RFC1952]  
4693 P. Deutsch, "GZIP file format specification version 4.3", RFC 1952, May 1996.
- 4694 [RFC2026]  
4695 S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.
- 4696 [RFC2045]  
4697 N. Fried, N. Borenstein, ", Multipurpose Internet Mail Extensions (MIME) Part One: Format of  
4698 Internet Message Bodies " RFC 2045, November 1996.
- 4699 [RFC2046]  
4700 Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. N. Freed & N. Borenstein.  
4701 November 1996. (Obsoletes RFC1521, RFC1522, RFC1590), RFC 2046.
- 4702 [RFC2048]  
4703 N. Freed, J. Klensin & J. Postel, "Multipurpose Internet Mail Extension (MIME) Part Four:  
4704 Registration Procedures". RFC 2048, November 1996.

- 4705 [RFC2068]  
4706 R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, "Hypertext Transfer Protocol -  
4707 HTTP/1.1", RFC 2068, January 1997
- 4708 [RFC2069]  
4709 J. Franks, P. Hallam-Baker, J. Hostetler, P. Leach, A. Luotonen, E. Sink, L. Stewart, "An Extension  
4710 to HTTP: Digest Access Authentication", RFC-2069, Jan 1997.
- 4711 [RFC2119]  
4712 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 , March 1997
- 4713 [RFC2228]  
4714 M. Horowitz, S. Lunt, "FTP Security Extensions", RFC 2228, October 1997.
- 4715 [RFC2246]  
4716 T. Dierks, C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.
- 4717 [RFC2277]  
4718 H. Alvestrand, "IETF Policy on Character Sets and Languages" RFC 2277, January 1998.
- 4719 [RFC2278]  
4720 N. Freed, J. Postel: "IANA CharSet Registration Procedures", RFC 2278, January 1998.
- 4721 [RFC2279]  
4722 F. Yergeau , "UTF-8, a transformation format of ISO 10646", RFC 2279. January 1998.
- 4723 [RFC2316]  
4724 S. Bellovin , "Report of the IAB Security Architecture Workshop", RFC 2316, April 1998.
- 4725 [RFC2396]  
4726 Berners-Lee, T., Fielding, R., Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax",  
4727 RFC 2396, August 1998.
- 4728 [RFC2565]  
4729 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.0: Encoding and  
4730 Transport", RFC 2565, April 1999.
- 4731 [RFC2566]  
4732 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and  
4733 Semantics", RFC 2566, April 1999.
- 4734 [RFC2567]  
4735 Wright, D., "Design Goals for an Internet Printing Protocol", draft-ietf-ipp-req-03.txt, November,  
4736 1998.

- 4737 [RFC2568]  
4738 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",  
4739 draft-ietf-ipp-rat-04.txt, November, 1998.
- 4740 [RFC2569]  
4741 Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", draft-  
4742 ietf-ipp-lpd-ipp-map-05.txt, November 1998.
- 4743 [RFC2579]  
4744 K. McCloghrie, D. Perkins, J. Schoenwaelder, "Textual Conventions for SMIV2" RFC 2579 (Also  
4745 STD0058), April 1999, .
- 4746 [SSL]  
4747 Netscape, The SSL Protocol, Version 3, (Text version 3.02), November 1996.
- 4748 [SWP]  
4749 P. Moore, B. Jahromi, S. Butler, "Simple Web Printing SWP/1.0", May 7, 1997,  
4750 [ftp://ftp.pwg.org/pub/pwg/ipp/new\\_PRO/swp9705.pdf](ftp://ftp.pwg.org/pub/pwg/ipp/new_PRO/swp9705.pdf)

4751 10. Author's Address

4752 Scott A. Isaacson (Editor)  
4753 Novell, Inc.  
4754 122 E 1700 S  
4755 Provo, UT 84606

4756  
4757 Phone: 801-861-7366  
4758 Fax: 801-861-2517  
4759 e-mail: [sisaacson@novell.com](mailto:sisaacson@novell.com)

4760  
4761 Tom Hastings  
4762 Xerox Corporation  
4763 737 Hawaii St. ESAE 231  
4764 El Segundo, CA 90245

4765  
4766 Phone: 310-333-6413  
4767 Fax: 310-333-5514  
4768 e-mail: [hastings@cp10.es.xerox.com](mailto:hastings@cp10.es.xerox.com)

4769  
4770 Robert Herriot  
4771 Xerox Corp.  
4772 3400 Hill View Ave, Building 1  
4773 Palo Alto, CA 94304  
4774

4775 Phone: 650-813-7696  
4776 Fax: 650-813-6860  
4777 e-mail: robert.herriot@pahv.xerox.com

4778  
4779 Roger deBry  
4780 Utah Valley State College  
4781 Orem, UT 84058

4782  
4783 Phone: (801) 222-8000  
4784 EMail: debryro@uvsc.edu

4785  
4786 Patrick Powell  
4787 Astart Technologies  
4788 9475 Chesapeake Dr., Suite D  
4789 San Diego, CA 95123

4790  
4791 Phone: (619) 874-6543  
4792 Fax: (619) 279-8424  
4793 e-mail: papowell@astart.com

4794  
4795 IPP Mailing List: [ipp@pwg.org](mailto:ipp@pwg.org)  
4796 IPP Mailing List Subscription: [ipp-request@pwg.org](mailto:ipp-request@pwg.org)  
4797 IPP Web Page: <http://www.pwg.org/ipp/>

4798  
4799 Implementers of this specification document are encouraged to join IPP Mailing List in order to participate  
4800 in any discussions of clarification issues and review of registration proposals for additional attributes and  
4801 values.

4802  
4803 Other Participants:

Chuck Adams - Tektronix  
Jeff Barnett - IBM  
Sylvan Butler - HP  
Angelo Caruso - Xerox  
Josh Cohen - Microsoft  
Andy Davidson - Tektronix  
Lee Farrell - Canon Information Systems  
Sue Gleeson - Digital  
Brian Grimshaw - Apple  
Richard Hart - Digital  
Zhi-Hong Huang - Zenographics  
Swen Johnson - Xerox  
Carl Kugler - IBM  
Takami Kurono - Brother  
Scott Lawrence - Agranot Systems  
Harry Lewis - IBM  
Roy Lomicka - Digital  
Ray Lutz - Cognisys  
Daniel Manchala - Xerox

Shivaun Albright - HP  
Ron Bergman - Dataproducts Corp.  
Keith Carter - IBM Corporation  
Rajesh Chawla - TR Computing Solutions  
Jeff Copeland - QMS  
Mabry Dozier - QMS  
Steve Gebert - IBM  
Charles Gordon - Osicom  
Jerry Hadsell - IBM  
Stephen Holmstead  
Babek Jahromi - Microsoft  
David Kellerman - Northlake Software  
Dave Kuntz - Hewlett-Packard  
Rick Landau - Digital  
Greg LeClair - Epson  
Tony Liao - Vivid Image  
Pete Loya - HP  
Mike MacKay - Novell, Inc.  
Carl-Uno Manros - Xerox

Jay Martin - Underscore  
Stan McConnell - Xerox  
Peter Michalek - Shinesoft  
Tetsuya Morita - Ricoh  
Pat Nogay - IBM  
Bob Pentecost - HP  
Xavier Riley - Xerox  
Gary Roberts - Ricoh  
Hiroyuki Sato - Canon  
Kris Schoff - HP  
Devon Taylor - Novell, Inc.  
Randy Turner - Sharp  
Jim Walker - DAZEL  
Rob Whittle - Novell, Inc.  
Don Wright - Lexmark  
Lloyd Young - Lexmark  
Peter Zehler - Xerox  
Steve Zilles - Adobe

Larry Masinter - Xerox  
Ira McDonald - High North Inc.  
Paul Moore - Microsoft  
Yuichi Niwa - Ricoh  
Ron Norton - Printronics  
Rob Rhoads - Intel  
David Roach - Unisys  
Stuart Rowley - Kyocera  
Richard Schneider - Epson  
Bob Setterbo - Adobe  
Mike Timperman - Lexmark  
Bill Wagner - Osicom  
Chris Wellens - Interworking Labs  
Jasper Wong - Xionics  
Rick Yardumian - Xerox  
Atsushi Yuki - Kyocera  
Frank Zhao - Panasonic

## 4804 11. Formats for IPP Registration Proposals

4805 In order to propose an IPP extension for registration, the proposer must submit an application to IANA by  
4806 email to "iana@iana.org" or by filling out the appropriate form on the IANA web pages  
4807 (<http://www.iana.org>). This section specifies the required information and the formats for proposing  
4808 registrations of extensions to IPP as provided in Section 6 for:

4809

- 4810 1. type2 'keyword' attribute values
- 4811 2. type3 'keyword' attribute values
- 4812 3. type2 'enum' attribute values
- 4813 4. type3 'enum' attribute values
- 4814 5. attributes
- 4815 6. attribute syntaxes
- 4816 7. operations
- 4817 8. status codes

## 4818 11.1 Type2 keyword attribute values registration

4819 Type of registration: type2 keyword attribute value

4820 Name of attribute to which this keyword specification is to be added:

4821 Proposed keyword name of this keyword value:

4822 Specification of this keyword value (follow the style of IPP Model Section 4.1.2.3):

4823 Name of proposer:

4824 Address of proposer:

4825 Email address of proposer:

4826

4827 Note: For type2 keywords, the Designated Expert will be the point of contact for the approved registration  
4828 specification, if any maintenance of the registration specification is needed.

## 4829 11.2 Type3 keyword attribute values registration

4830 Type of registration: type3 keyword attribute value

4831 Name of attribute to which this keyword specification is to be added:

4832 Proposed keyword name of this keyword value:

4833 Specification of this keyword value (follow the style of IPP Model Section 4.1.2.3):

4834 Name of proposer:

4835 Address of proposer:

4836 Email address of proposer:

4837

4838 Note: For type3 keywords, the proposer will be the point of contact for the approved registration  
4839 specification, if any maintenance of the registration specification is needed.

## 4840 11.3 Type2 enum attribute values registration

4841 Type of registration: type2 enum attribute value

4842 Name of attribute to which this enum specification is to be added:

4843 Keyword symbolic name of this enum value:

4844 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

4845 Specification of this enum value (follow the style of IPP Model Section 4.1.4):

4846 Name of proposer:

4847 Address of proposer:

4848 Email address of proposer:

4849

4850 Note: For type2 enums, the Designated Expert will be the point of contact for the approved registration  
4851 specification, if any maintenance of the registration specification is needed.

#### 4852 11.4 Type3 enum attribute values registration

4853 Type of registration: type3 enum attribute value

4854 Name of attribute to which this enum specification is to be added:

4855 Keyword symbolic name of this enum value:

4856 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

4857 Specification of this enum value (follow the style of IPP Model Section 4.1.4):

4858 Name of proposer:

4859 Address of proposer:

4860 Email address of proposer:

4861

4862 Note: For type3 enums, the proposer will be the point of contact for the approved registration specification,  
4863 if any maintenance of the registration specification is needed.

#### 4864 11.5 Attribute registration

4865 Type of registration: attribute

4866 Proposed keyword name of this attribute:

4867 Types of attribute (Operation, Job Template, Job Description, Printer Description):

4868 Operations to be used with if the attribute is an operation attribute:

4869 Object (Job, Printer, etc. if bound to an object):

4870 Attribute syntax(es) (include 1setOf and range as in Section 4.2):

4871 If attribute syntax is 'keyword' or 'enum', is it type2 or type3:

4872 If this is a Printer attribute, MAY the value returned depend on "document-format" (See Section 6.2):

4873 If this is a Job Template attribute, how does its specification depend on the value of the "multiple-  
4874 document-handling" attribute:

4875 Specification of this attribute (follow the style of IPP Model Section 4.2):

4876 Name of proposer:

4877 Address of proposer:

4878 Email address of proposer:

4879

4880 Note: For attributes, the IPP Designated Expert will be the point of contact for the approved registration  
4881 specification, if any maintenance of the registration specification is needed.



## 4882 11.6 Attribute Syntax registration

4883 Type of registration: attribute syntax

4884 Proposed name of this attribute syntax:

4885 Type of attribute syntax (integer, octetString, character-string, see [IPP-PRO]):

4886 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

4887 Specification of this attribute (follow the style of IPP Model Section 4.1):

4888 Name of proposer:

4889 Address of proposer:

4890 Email address of proposer:

4891

4892 Note: For attribute syntaxes, the IPP Designated Expert will be the point of contact for the approved  
4893 registration specification, if any maintenance of the registration specification is needed.

## 4894 11.7 Operation registration

4895 Type of registration: operation

4896 Proposed name of this operation:

4897 Numeric operation-id value (to be assigned by the IPP Designated Expert in consultation with IANA):

4898 Object Target (Job, Printer, etc. that operation is upon):

4899 Specification of this attribute (follow the style of IPP Model Section 3):

4900 Name of proposer:

4901 Address of proposer:

4902 Email address of proposer:

4903

4904 Note: For operations, the IPP Designated Expert will be the point of contact for the approved registration  
4905 specification, if any maintenance of the registration specification is needed.

## 4906 11.8 Attribute Group registration

4907 Type of registration: attribute group

4908 Proposed name of this attribute group:

4909 Numeric tag according to [IPP-PRO] (to be assigned by the IPP Designated Expert in consultation with  
4910 IANA):

4911 Operation requests and group number for each operation in which the attribute group occurs:

4912 Operation responses and group number for each operation in which the attribute group occurs:

4913 Specification of this attribute group (follow the style of IPP Model Section 3):

4914 Name of proposer:

4915 Address of proposer:

4916 Email address of proposer:

4917

4918 Note: For attribute groups, the IPP Designated Expert will be the point of contact for the approved  
4919 registration specification, if any maintenance of the registration specification is needed.

## 4920 11.9 Status code registration

4921 Type of registration: status code

4922 Keyword symbolic name of this status code value:

4923 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

4924 Operations that this status code may be used with:

4925 Specification of this status code (follow the style of IPP Model Section 13 APPENDIX B: Status Codes  
4926 and Suggested Status Code Messages):

4927 Name of proposer:

4928 Address of proposer:

4929 Email address of proposer:

4930

4931 Note: For status codes, the Designated Expert will be the point of contact for the approved registration  
4932 specification, if any maintenance of the registration specification is needed.

## 4933 12. APPENDIX A: Terminology

4934 This specification document uses the terminology defined in this section.

## 4935 12.1 Conformance Terminology

4936 The key words "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT",

4937 "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in

4938 RFC 2119 [RFC2119].

## 4939 12.1.1 NEED NOT

4940 This term is not included in RFC 2119. The verb "NEED NOT" indicates an action that the subject of the

4941 sentence does not have to implement in order to claim conformance to the standard. The verb "NEED

4942 NOT" is used instead of "MAY NOT" since "MAY NOT" sounds like a prohibition.

## 4943 12.2 Model Terminology

## 4944 12.2.1 Keyword

4945 Keywords are used within this document as identifiers of semantic entities within the abstract model (see

4946 section 4.1.2.3). Attribute names, some attribute values, attribute syntaxes, and attribute group names are

4947 represented as keywords.

## 4948 12.2.2 Attributes

4949 An attribute is an item of information that is associated with an instance of an IPP object. An attribute  
4950 consists of an attribute name and one or more attribute values. Each attribute has a specific attribute syntax.  
4951 All object attributes are defined in section 4 and all operation attributes are defined in section 3.

4952 Job Template Attributes are described in section 4.2. The client optionally supplies Job Template attributes  
4953 in a create request (operation requests that create Job objects). The Printer object has associated attributes  
4954 which define supported and default values for the Printer.

## 4955 12.2.2.1 Attribute Name

4956 Each attribute is uniquely identified in this document by its attribute name. An attribute name is a keyword.  
4957 The keyword attribute name is given in the section header describing that attribute. In running text in this  
4958 document, attribute names are indicated inside double quotation marks (") where the quotation marks are  
4959 not part of the keyword itself.

## 4960 12.2.2.2 Attribute Group Name

4961 Related attributes are grouped into named groups. The name of the group is a keyword. The group name  
4962 may be used in place of naming all the attributes in the group explicitly. Attribute groups are defined in  
4963 section 3.

## 4964 12.2.2.3 Attribute Value

4965 Each attribute has one or more values. Attribute values are represented in the syntax type specified for that  
4966 attribute. In running text in this document, attribute values are indicated inside single quotation marks ('),  
4967 whether their attribute syntax is keyword, integer, text, etc. where the quotation marks are not part of the  
4968 value itself.

## 4969 12.2.2.4 Attribute Syntax

4970 Each attribute is defined using an explicit syntax type. In this document, each syntax type is defined as a  
4971 keyword with specific meaning. The "Encoding and Transport" document [IPP-PRO] indicates the actual  
4972 "on-the-wire" encoding rules for each syntax type. Attribute syntax types are defined in section 4.1.

## 4973 12.2.3 Supports

4974 By definition, a Printer object supports an attribute only if that Printer object responds with the  
4975 corresponding attribute populated with some value(s) in a response to a query for that attribute. A Printer  
4976 object supports an attribute value if the value is one of the Printer object's "supported values" attributes.  
4977 The device behind a Printer object may exhibit a behavior that corresponds to some IPP attribute, but if the  
4978 Printer object, when queried for that attribute, doesn't respond with the attribute, then as far as IPP is  
4979 concerned, that implementation does not support that feature. If the Printer object's "xxx-supported"

4980 attribute is not populated with a particular value (even if that value is a legal value for that attribute), then  
4981 that Printer object does not support that particular value.

4982 A conforming implementation MUST support all REQUIRED attributes. However, even for REQUIRED  
4983 attributes, conformance to IPP does not mandate that all implementations support all possible values  
4984 representing all possible job processing behaviors and features. For example, if a given instance of a  
4985 Printer supports only certain document formats, then that Printer responds with the "document-format-  
4986 supported" attribute populated with a set of values, possibly only one, taken from the entire set of possible  
4987 values defined for that attribute. This limited set of values represents the Printer's set of supported  
4988 document formats. Supporting an attribute and some set of values for that attribute enables IPP end users to  
4989 be aware of and make use of those features associated with that attribute and those values. If an  
4990 implementation chooses to not support an attribute or some specific value, then IPP end users would have  
4991 no ability to make use of that feature within the context of IPP itself. However, due to existing practice and  
4992 legacy systems which are not IPP aware, there might be some other mechanism outside the scope of IPP to  
4993 control or request the "unsupported" feature (such as embedded instructions within the document data  
4994 itself).

4995 For example, consider the "finishings-supported" attribute.

- 4996 1) If a Printer object is not physically capable of stapling, the "finishings-supported" attribute MUST  
4997 NOT be populated with the value of 'staple'.
- 4998 2) A Printer object is physically capable of stapling, however an implementation chooses not to support  
4999 stapling in the IPP "finishings" attribute. In this case, 'staple' MUST NOT be a value in the  
5000 "finishings-supported" Printer object attribute. Without support for the value 'staple', an IPP end  
5001 user would have no means within the protocol itself to request that a Job be stapled. However, an  
5002 existing document data formatter might be able to request that the document be stapled directly with  
5003 an embedded instruction within the document data. In this case, the IPP implementation does not  
5004 "support" stapling, however the end user is still able to have some control over the stapling of the  
5005 completed job.
- 5006 3) A Printer object is physically capable of stapling, and an implementation chooses to support stapling  
5007 in the IPP "finishings" attribute. In this case, 'staple' MUST be a value in the "finishings-supported"  
5008 Printer object attribute. Doing so, would enable end users to be aware of and make use of the  
5009 stapling feature using IPP attributes.

5010

5011 Even though support for Job Template attributes by a Printer object is OPTIONAL, it is RECOMMENDED  
5012 that if the device behind a Printer object is capable of realizing any feature or function that corresponds to  
5013 an IPP attribute and some associated value, then that implementation SHOULD support that IPP attribute  
5014 and value.

5015 The set of values in any of the supported value attributes is set (populated) by some administrative process  
5016 or automatic sensing mechanism that is outside the scope of this IPP/1.1 document. For administrative  
5017 policy and control reasons, an administrator may choose to make only a subset of possible values visible to  
5018 the end user. In this case, the real output device behind the IPP Printer abstraction may be capable of a  
5019 certain feature, however an administrator is specifying that access to that feature not be exposed to the end  
5020 user through the IPP protocol. Also, since a Printer object may represent a logical print device (not just a  
5021 physical device) the actual process for supporting a value is undefined and left up to the implementation.

5022 However, if a Printer object supports a value, some manual human action may be needed to realize the  
5023 semantic action associated with the value, but no end user action is required.

5024 For example, if one of the values in the "finishings-supported" attribute is 'staple', the actual process might  
5025 be an automatic staple action by a physical device controlled by some command sent to the device. Or, the  
5026 actual process of stapling might be a manual action by an operator at an operator attended Printer object.

5027 For another example of how supported attributes function, consider a system administrator who desires to  
5028 control all print jobs so that no job sheets are printed in order to conserve paper. To force no job sheets, the  
5029 system administrator sets the only supported value for the "job-sheets-supported" attribute to 'none'. In this  
5030 case, if a client requests anything except 'none', the create request is rejected or the "job-sheets" value is  
5031 ignored (depending on the value of "ipp-attribute-fidelity"). To force the use of job start/end sheets on all  
5032 jobs, the administrator does not include the value 'none' in the "job-sheets-supported" attribute. In this case,  
5033 if a client requests 'none', the create request is rejected or the "job-sheets" value is ignored (again depending  
5034 on the value of "ipp-attribute-fidelity").

#### 5035 12.2.4 print-stream page

5036 A "print-stream page" is a page according to the definition of pages in the language used to express the  
5037 document data.

#### 5038 12.2.5 impression

5039 An "impression" is the image (possibly many print-stream pages in different configurations) imposed onto a  
5040 single media page.

### 5041 13. APPENDIX B: Status Codes and Suggested Status Code Messages

5042 This section defines status code enum keywords and values that are used to provide semantic information  
5043 on the results of an operation request. Each operation response **MUST** include a status code. The response  
5044 **MAY** also contain a status message that provides a short textual description of the status. The status code  
5045 is intended for use by automata, and the status message is intended for the human end user. Since the status  
5046 message is an **OPTIONAL** component of the operation response, an IPP application (i.e., a browser, GUI,  
5047 print driver or gateway) is **NOT REQUIRED** to examine or display the status message, since it **MAY** not be  
5048 returned to the application.

5049 The prefix of the status keyword defines the class of response as follows:

- 5050 "informational" - Request received, continuing process
- 5051 "successful" - The action was successfully received, understood, and accepted
- 5052 "redirection" - Further action must be taken in order to complete the request
- 5053 "client-error" - The request contains bad syntax or cannot be fulfilled
- 5054 "server-error" - The IPP object failed to fulfill an apparently valid request

5055

5056 As with type2 enums, IPP status codes are extensible. IPP clients are NOT REQUIRED to understand the  
5057 meaning of all registered status codes, though such understanding is obviously desirable. However, IPP  
5058 clients MUST understand the class of any status code, as indicated by the prefix, and treat any unrecognized  
5059 response as being equivalent to the first status code of that class, with the exception that an unrecognized  
5060 response MUST NOT be cached. For example, if an unrecognized status code of "client-error-xxx-yyy" is  
5061 received by the client, it can safely assume that there was something wrong with its request and treat the  
5062 response as if it had received a "client-error-bad-request" status code. In such cases, IPP applications  
5063 SHOULD present the OPTIONAL message (if present) to the end user since the message is likely to  
5064 contain human readable information which will help to explain the unusual status. The name of the enum  
5065 is the suggested status message for US English.

5066 The status code values range from 0x0000 to 0x7FFF. The value ranges for each status code class are as  
5067 follows:

5068 "successful" - 0x0000 to 0x00FF  
5069 "informational" - 0x0100 to 0x01FF  
5070 "redirection" - 0x0200 to 0x02FF  
5071 "client-error" - 0x0400 to 0x04FF  
5072 "server-error" - 0x0500 to 0x05FF  
5073

5074 The top half (128 values) of each range (0x0n40 to 0x0nFF, for n = 0 to 5) is reserved for private use within  
5075 each status code class. Values 0x0600 to 0x7FFF are reserved for future assignment and MUST NOT be  
5076 used.

## 5077 13.1 Status Codes

5078 Each status code is described below. Section 13.1.5.9 contains a table that indicates which status codes  
5079 apply to which operations. The Implementer's Guide [IPP-IIG] describe the suggested steps for processing  
5080 IPP attributes for all operations, including returning status codes.

### 5081 13.1.1 Informational

5082 This class of status code indicates a provisional response and is to be used for informational purposes only.

5083 There are no status codes defined in IPP/1.1 for this class of status code.

### 5084 13.1.2 Successful Status Codes

5085 This class of status code indicates that the client's request was successfully received, understood, and  
5086 accepted.

#### 5087 13.1.2.1 successful-ok (0x0000)

5088 The request has succeeded and no request attributes were substituted or ignored. In the case of a response  
5089 to a create request, the 'successful-ok' status code indicates that the request was successfully received and

5090 validated, and that the Job object has been created; it does not indicate that the job has been processed. The  
5091 transition of the Job object into the 'completed' state is the only indicator that the job has been printed.

#### 5092 13.1.2.2 successful-ok-ignored-or-substituted-attributes (0x0001)

5093 The request has succeeded, but some supplied (1) attributes were ignored or (2) unsupported values were  
5094 substituted with supported values or were ignored in order to perform the operation without rejecting it.  
5095 Unsupported attributes, attribute syntaxes, or values **MUST** be returned in the Unsupported Attributes  
5096 group of the response for all operations. There is an exception to this rule for the query operations: Get-  
5097 Printer-Attributes, Get-Jobs, and Get-Job-Attributes for the "requested-attributes" operation attribute only.  
5098 When the supplied values of the "requested-attributes" operation attribute are requesting attributes that are  
5099 not supported, the IPP object **MAY**, but is **NOT REQUIRED** to, return the "requested-attributes" attribute  
5100 in the Unsupported Attribute response group (with the unsupported values only). See sections 3.1.7 and  
5101 3.2.1.2.

#### 5102 13.1.2.3 successful-ok-conflicting-attributes (0x0002)

5103 The request has succeeded, but some supplied attribute values conflicted with the values of other supplied  
5104 attributes. These conflicting values were either (1) substituted with (supported) values or (2) the attributes  
5105 were removed in order to process the job without rejecting it. Attributes or values which conflict with other  
5106 attributes and have been substituted or ignored **MUST** be returned in the Unsupported Attributes group of  
5107 the response for all operations as supplied by the client. See sections 3.1.7 and 3.2.1.2.

#### 5108 13.1.3 Redirection Status Codes

5109 This class of status code indicates that further action needs to be taken to fulfill the request.

5110 There are no status codes defined in IPP/1.1 for this class of status code.

#### 5111 13.1.4 Client Error Status Codes

5112 This class of status code is intended for cases in which the client seems to have erred. The IPP object  
5113 **SHOULD** return a message containing an explanation of the error situation and whether it is a temporary or  
5114 permanent condition.

#### 5115 13.1.4.1 client-error-bad-request (0x0400)

5116 The request could not be understood by the IPP object due to malformed syntax (such as the value of a  
5117 fixed length attribute whose length does not match the prescribed length for that attribute - see the  
5118 Implementer's Guide [IPP-IIG]). The IPP application **SHOULD NOT** repeat the request without  
5119 modifications.

## 5120 13.1.4.2 client-error-forbidden (0x0401)

5121 The IPP object understood the request, but is refusing to fulfill it. Additional authentication information or  
5122 authorization credentials will not help and the request SHOULD NOT be repeated. This status code is  
5123 commonly used when the IPP object does not wish to reveal exactly why the request has been refused or  
5124 when no other response is applicable.

## 5125 13.1.4.3 client-error-not-authenticated (0x0402)

5126 The request requires user authentication. The IPP client may repeat the request with suitable authentication  
5127 information. If the request already included authentication information, then this status code indicates that  
5128 authorization has been refused for those credentials. If this response contains the same challenge as the  
5129 prior response, and the user agent has already attempted authentication at least once, then the response  
5130 message may contain relevant diagnostic information. This status codes reveals more information than  
5131 "client-error-forbidden".

## 5132 13.1.4.4 client-error-not-authorized (0x0403)

5133 The requester is not authorized to perform the request. Additional authentication information or  
5134 authorization credentials will not help and the request SHOULD NOT be repeated. This status code is used  
5135 when the IPP object wishes to reveal that the authentication information is understandable, however, the  
5136 requester is explicitly not authorized to perform the request. This status codes reveals more information  
5137 than "client-error-forbidden" and "client-error-not-authenticated".

## 5138 13.1.4.5 client-error-not-possible (0x0404)

5139 This status code is used when the request is for something that can not happen. For example, there might  
5140 be a request to cancel a job that has already been canceled or aborted by the system. The IPP client  
5141 SHOULD NOT repeat the request.

## 5142 13.1.4.6 client-error-timeout (0x0405)

5143 The client did not produce a request within the time that the IPP object was prepared to wait. For example,  
5144 a client issued a Create-Job operation and then, after a long period of time, issued a Send-Document  
5145 operation and this error status code was returned in response to the Send-Document request (see section  
5146 3.3.1). The IPP object might have been forced to clean up resources that had been held for the waiting  
5147 additional Documents. The IPP object was forced to close the Job since the client took too long. The client  
5148 SHOULD NOT repeat the request without modifications.

## 5149 13.1.4.7 client-error-not-found (0x0406)

5150 The IPP object has not found anything matching the request URI. No indication is given of whether the  
5151 condition is temporary or permanent. For example, a client with an old reference to a Job (a URI) tries to  
5152 cancel the Job, however in the mean time the Job might have been completed and all record of it at the  
5153 Printer has been deleted. This status code, 'client-error-not-found' is returned indicating that the referenced



5154 Job can not be found. This error status code is also used when a client supplies a URI as a reference to the  
5155 document data in either a Print-URI or Send-URI operation, but the document can not be found.

5156 In practice, an IPP application should avoid a not found situation by first querying and presenting a list of  
5157 valid Printer URIs and Job URIs to the end-user.

#### 5158 13.1.4.8 client-error-gone (0x0407)

5159 The requested object is no longer available and no forwarding address is known. This condition should be  
5160 considered permanent. Clients with link editing capabilities should delete references to the request URI  
5161 after user approval. If the IPP object does not know or has no facility to determine, whether or not the  
5162 condition is permanent, the status code "client-error-not-found" should be used instead.

5163 This response is primarily intended to assist the task of maintenance by notifying the recipient that the  
5164 resource is intentionally unavailable and that the IPP object administrator desires that remote links to that  
5165 resource be removed. It is not necessary to mark all permanently unavailable resources as "gone" or to keep  
5166 the mark for any length of time -- that is left to the discretion of the IPP object administrator.

#### 5167 13.1.4.9 client-error-request-entity-too-large (0x0408)

5168 The IPP object is refusing to process a request because the request entity is larger than the IPP object is  
5169 willing or able to process. An IPP Printer returns this status code when it limits the size of print jobs and it  
5170 receives a print job that exceeds that limit or when the attributes are so many that their encoding causes the  
5171 request entity to exceed IPP object capacity.

#### 5172 13.1.4.10 client-error-request-value-too-long (0x0409)

5173 The IPP object is refusing to service the request because one or more of the client-supplied attributes has a  
5174 variable length value that is longer than the maximum length specified for that attribute. The IPP object  
5175 might not have sufficient resources (memory, buffers, etc.) to process (even temporarily), interpret, and/or  
5176 ignore a value larger than the maximum length. Another use of this error code is when the IPP object  
5177 supports the processing of a large value that is less than the maximum length, but during the processing of  
5178 the request as a whole, the object may pass the value onto some other system component which is not able  
5179 to accept the large value. For more details, see the Implementer's Guide [IPP-IIG] .

5180 Note: For attribute values that are URIs, this rare condition is only likely to occur when a client has  
5181 improperly submitted a request with long query information (e.g. an IPP application allows an end-user to  
5182 enter an invalid URI), when the client has descended into a URI "black hole" of redirection (e.g., a  
5183 redirected URI prefix that points to a suffix of itself), or when the IPP object is under attack by a client  
5184 attempting to exploit security holes present in some IPP objects using fixed-length buffers for reading or  
5185 manipulating the Request-URI.

## 5186 13.1.4.11 client-error-document-format-not-supported (0x040A)

5187 The IPP object is refusing to service the request because the document data is in a format, as specified in  
5188 the "document-format" operation attribute, that is not supported by the Printer object. This error is returned  
5189 independent of the client-supplied "ipp-attribute-fidelity". The Printer object MUST return this status code,  
5190 even if there are other Job Template attributes that are not supported as well, since this error is a bigger  
5191 problem than with Job Template attributes. See sections 3.1.7 and 3.2.1.1.

## 5192 13.1.4.12 client-error-attributes-or-values-not-supported (0x040B)

5193 In a create request, if the Printer object does not support one or more attributes, attribute syntaxes, or  
5194 attribute values supplied in the request and the client supplied the "ipp-attributes-fidelity" operation  
5195 attribute with the 'true' value, the Printer object MUST return this status code. The Printer object MUST  
5196 also return in the Unsupported Attributes Group all the attributes and/or values supplied by the client that  
5197 are not supported. See section 3.1.7. For example, if the request indicates 'iso-a4' media, but that media  
5198 type is not supported by the Printer object. Or, if the client supplies a Job Template attribute and the  
5199 attribute itself is not even supported by the Printer. If the "ipp-attribute-fidelity" attribute is 'false', the  
5200 Printer MUST ignore or substitute values for unsupported Job Template attributes and values rather than  
5201 reject the request and return this status code.

5202 For any operation where a client requests attributes (such as a Get-Jobs, Get-Printer-Attributes, or Get-Job-  
5203 Attributes operation), if the IPP object does not support one or more of the requested attributes, the IPP  
5204 object simply ignores the unsupported requested attributes and processes the request as if they had not been  
5205 supplied, rather than returning this status code. In this case, the IPP object MUST return the 'successful-ok-  
5206 ignored-or-substituted-attributes' status code and MAY return the unsupported attributes as values of the  
5207 "requested-attributes" in the Unsupported Attributes Group (see section 13.1.2.2).

## 5208 13.1.4.13 client-error-uri-scheme-not-supported (0x040C)

5209 The scheme of the client-supplied URI in a Print-URI or a Send-URI operation is not supported. See  
5210 section 3.1.7.

## 5211 13.1.4.14 client-error-charset-not-supported (0x040D)

5212 For any operation, if the IPP Printer does not support the charset supplied by the client in the "attributes-  
5213 charset" operation attribute, the Printer MUST reject the operation and return this status and any 'text' or  
5214 'name' attributes using the 'utf-8' charset (see Section 3.1.4.1). See section 3.1.7.

## 5215 13.1.4.15 client-error-conflicting-attributes (0x040E)

5216 The request is rejected because some attribute values conflicted with the values of other attributes which  
5217 this document does not permit to be substituted or ignored. The Printer object MUST also return in the  
5218 Unsupported Attributes Group the conflicting attributes supplied by the client. See sections 3.1.7 and  
5219 3.2.1.2.

## 5220 13.1.4.16 client-error-compression-not-supported (0x040F)

5221 The IPP object is refusing to service the request because the document data, as specified in the  
5222 "compression" operation attribute, is compressed in a way that is not supported by the Printer object. This  
5223 error is returned independent of the client-supplied "ipp-attribute-fidelity". The Printer object MUST return  
5224 this status code, even if there are other Job Template attributes that are not supported as well, since this  
5225 error is a bigger problem than with Job Template attributes. See sections 3.1.7 and 3.2.1.1.

## 5226 13.1.4.17 client-error-compression-error (0x0410)

5227 The IPP object is refusing to service the request because the document data cannot be decompressed when  
5228 using the algorithm specified by the "compression" operation attribute. This error is returned independent  
5229 of the client-supplied "ipp-attribute-fidelity". The Printer object MUST return this status code, even if there  
5230 are Job Template attributes that are not supported as well, since this error is a bigger problem than with Job  
5231 Template attributes. See sections 3.1.7 and 3.2.1.1.

## 5232 13.1.4.18 client-error-document-format-error (0x0411)

5233 The IPP object is refusing to service the request because Printer encountered an error in the document data  
5234 while interpreting it. This error is returned independent of the client-supplied "ipp-attribute-fidelity". The  
5235 Printer object MUST return this status code, even if there are Job Template attributes that are not supported  
5236 as well, since this error is a bigger problem than with Job Template attributes. See sections 3.1.7 and  
5237 3.2.1.1.

## 5238 13.1.4.19 client-error-document-access-error (0x0412)

5239 The IPP object is refusing to service the Print-URI or Send-URI request because Printer encountered an  
5240 access error while attempting to validate the accessibility or access the document data specified in the  
5241 "document-uri" operation attribute. The Printer MAY also return a specific document access error code  
5242 using the "document-access-error" operation attribute (see section 3.1.6.4). This error is returned  
5243 independent of the client-supplied "ipp-attribute-fidelity". The Printer object MUST return this status code,  
5244 even if there are Job Template attributes that are not supported as well, since this error is a bigger problem  
5245 than with Job Template attributes. See section 3.1.7.

## 5246 13.1.5 Server Error Status Codes

5247 This class of status codes indicates cases in which the IPP object is aware that it has erred or is incapable of  
5248 performing the request. The IPP object SHOULD include a message containing an explanation of the error  
5249 situation, and whether it is a temporary or permanent condition.

## 5250 13.1.5.1 server-error-internal-error (0x0500)

5251 The IPP object encountered an unexpected condition that prevented it from fulfilling the request. This error  
5252 status code differs from "server-error-temporary-error" in that it implies a more permanent type of internal  
5253 error. It also differs from "server-error-device-error" in that it implies an unexpected condition (unlike a

5254 paper-jam or out-of-toner problem which is undesirable but expected). This error status code indicates that  
5255 probably some knowledgeable human intervention is required.

#### 5256 13.1.5.2 server-error-operation-not-supported (0x0501)

5257 The IPP object does not support the functionality required to fulfill the request. This is the appropriate  
5258 response when the IPP object does not recognize an operation or is not capable of supporting it. See section  
5259 3.1.7.

#### 5260 13.1.5.3 server-error-service-unavailable (0x0502)

5261 The IPP object is currently unable to handle the request due to a temporary overloading or maintenance of  
5262 the IPP object. The implication is that this is a temporary condition which will be alleviated after some  
5263 delay. If known, the length of the delay may be indicated in the message. If no delay is given, the IPP  
5264 application should handle the response as it would for a "server-error-temporary-error" response. If the  
5265 condition is more permanent, the error status codes "client-error-gone" or "client-error-not-found" could be  
5266 used.

#### 5267 13.1.5.4 server-error-version-not-supported (0x0503)

5268 The IPP object does not support, or refuses to support, the IPP protocol version that was supplied as the  
5269 value of the "version-number" operation parameter in the request. The IPP object is indicating that it is  
5270 unable or unwilling to complete the request using the same major and minor version number as supplied in  
5271 the request other than with this error message. The error response SHOULD contain a "status-message"  
5272 attribute (see section 3.1.6.2) describing why that version is not supported and what other versions are  
5273 supported by that IPP object. See section 3.1.8.

5274 The error response MUST identify in the "version-number" operation parameter the closest version number  
5275 that the IPP object does support. For example, if a client supplies version '1.0' and an IPP/1.1 object  
5276 supports version '1.0', then it MUST respond with version '1.0' in all responses to such a request. If the  
5277 IPP/1.1 object does not support version '1.0', then it SHOULD accept the request and respond with version  
5278 '1.1' or MAY reject the request and respond with this error code and version '1.1'. If a client supplies a  
5279 version '1.2' the IPP/1.1 object SHOULD accept the request and return version '1.1' or MAY reject the  
5280 request and respond with this error code and version '1.1'. See sections 3.1.8 and 4.4.14.

#### 5281 13.1.5.5 server-error-device-error (0x0504)

5282 A printer error, such as a paper jam, occurs while the IPP object processes a Print or Send operation. The  
5283 response contains the true Job Status (the values of the "job-state" and "job-state-reasons" attributes).  
5284 Additional information can be returned in the OPTIONAL "job-state-message" attribute value or in the  
5285 OPTIONAL status message that describes the error in more detail. This error status code is only returned in  
5286 situations where the Printer is unable to accept the create request because of such a device error. For  
5287 example, if the Printer is unable to spool, and can only accept one job at a time, the reason it might reject a  
5288 create request is that the printer currently has a paper jam. In many cases however, where the Printer object  
5289 can accept the request even though the Printer has some error condition, the 'successful-ok' status code will

5290 be returned. In such a case, the client would look at the returned Job Object Attributes or later query the  
5291 Printer to determine its state and state reasons.

#### 5292 13.1.5.6 server-error-temporary-error (0x0505)

5293 A temporary error such as a buffer full write error, a memory overflow (i.e. the document data exceeds the  
5294 memory of the Printer), or a disk full condition, occurs while the IPP Printer processes an operation. The  
5295 client MAY try the unmodified request again at some later point in time with an expectation that the  
5296 temporary internal error condition may have been cleared. Alternatively, as an implementation option, a  
5297 Printer object MAY delay the response until the temporary condition is cleared so that no error is returned.

#### 5298 13.1.5.7 server-error-not-accepting-jobs (0x0506)

5299 A temporary error indicating that the Printer is not currently accepting jobs, because the administrator has  
5300 set the value of the Printer's "printer-is-not-accepting-jobs" attribute to 'false' (by means outside the scope of  
5301 this IPP/1.1 document).

#### 5302 13.1.5.8 server-error-busy (0x0507)

5303 A temporary error indicating that the Printer is too busy processing jobs and/or other requests. The client  
5304 SHOULD try the unmodified request again at some later point in time with an expectation that the  
5305 temporary busy condition will have been cleared.

#### 5306 13.1.5.9 server-error-job-canceled (0x0508)

5307 An error indicating that the job has been canceled by an operator or the system while the client was  
5308 transmitting the data to the IPP Printer. If a job-id and job-uri had been created, then they are returned in  
5309 the Print-Job, Send-Document, or Send-URI response as usual; otherwise, no job-id and job-uri are returned  
5310 in the response.

#### 5311 13.1.5.10 server-error-multiple-document-jobs-not-supported (0x0509)

5312 The IPP object does not support multiple documents per job and a client attempted to supply document data  
5313 with a second Send-Document or Send-URI operation.

## 5314 13.2 Status Codes for IPP Operations

5315 PJ = Print-Job, PU = Print-URI, CJ = Create-Job, SD = Send-Document  
 5316 SU = Send-URI, V = Validate-Job, GA = Get-Job-Attributes and  
 5317 Get-Printer-Attributes, GJ = Get-Jobs, C = Cancel-Job

5318		IPP Operations								
5319	IPP Status Keyword	PJ	PU	CJ	SD	SU	V	GA	GJ	C
5320	-----	--	--	--	--	--	--	--	--	--
5321	successful-ok	x	x	x	x	x	x	x	x	x
5322	successful-ok-ignored-or-substituted-	x	x	x	x	x	x	x	x	x
5323	attributes									
5324	successful-ok-conflicting-attributes	x	x	x	x	x	x	x	x	x
5325	client-error-bad-request	x	x	x	x	x	x	x	x	x
5326	client-error-forbidden	x	x	x	x	x	x	x	x	x
5327	client-error-not-authenticated	x	x	x	x	x	x	x	x	x
5328	client-error-not-authorized	x	x	x	x	x	x	x	x	x
5329	client-error-not-possible	x	x	x	x	x	x	x	x	x
5330	client-error-timeout				x	x				
5331	client-error-not-found	x	x	x	x	x	x	x	x	x
5332	client-error-gone	x	x	x	x	x	x	x	x	x
5333	client-error-request-entity-too-large	x	x	x	x	x	x	x	x	x
5334	client-error-request-value-too-long	x	x	x	x	x	x	x	x	x
5335	client-error-document-format-not-	x	x		x	x	x	x		
5336	supported									
5337	client-error-attributes-or-values-not-	x	x	x	x	x	x	x	x	x
5338	supported									
5339	client-error-uri-scheme-not-supported		x			x				
5340	client-error-charset-not-supported	x	x	x	x	x	x	x	x	x
5341	client-error-conflicting-attributes	x	x	x	x	x	x	x	x	x
5342	client-error-compression-not-supported	x	x		x	x	x			
5343	client-error-compression-error	x	x		x	x				
5344	client-error-document-format-error	x	x		x	x				
5345	client-error-document-access-error		x			x				
5346	server-error-internal-error	x	x	x	x	x	x	x	x	x
5347	server-error-operation-not-supported		x	x	x	x				
5348	server-error-service-unavailable	x	x	x	x	x	x	x	x	x
5349	server-error-version-not-supported	x	x	x	x	x	x	x	x	x
5350	server-error-device-error	x	x	x	x	x				
5351	server-error-temporary-error	x	x	x	x	x				
5352	server-error-not-accepting-jobs	x	x	x			x			
5353	server-error-busy	x	x	x	x	x	x	x	x	x
5354	server-error-job-canceled	x			x	x				
5355	server-error-multiple-document-jobs-				x	x				
5356	not-supported									
5357										

5358 HJ = Hold-Job, RJ = Release-Job, RS = Restart-Job  
 5359 PP = Pause-Printer, RP = Resume-Printer, PJ = Purge-Jobs

5360

5361

## IPP Operations (cont.)

5362 IPP Status Keyword

HJ RJ RS PP RP PJ

5363 -----

-- -- -- -- -- --

5364 successful-ok

x x x x x x

5365 successful-ok-ignored-or-substituted-  
5366 attributes

x x x x x x

5367 successful-ok-conflicting-attributes

x x x x x x

5368 client-error-bad-request

x x x x x x

5369 client-error-forbidden

x x x x x x

5370 client-error-not-authenticated

x x x x x x

5371 client-error-not-authorized

x x x x x x

5372 client-error-not-possible

x x x x x x

5373 client-error-timeout

5374 client-error-not-found

x x x x x x

5375 client-error-gone

x x x x x x

5376 client-error-request-entity-too-large

x x x x x x

5377 client-error-request-value-too-long

x x x x x x

5378 client-error-document-format-not-  
5379 supported

x x x x x x

5380 client-error-attributes-or-values-not-  
5381 supported

x x x x x x

5382 client-error-uri-scheme-not-supported

5383 client-error-charset-not-supported

x x x x x x

5384 client-error-conflicting-attributes

x x x x x x

5385 client-error-compression-not-supported

5386 client-error-compression-error

5387 client-error-document-format-error

5388 client-error-document-access-error

5389 server-error-internal-error

x x x x x x

5390 server-error-operation-not-supported

x x x x x x

5391 server-error-service-unavailable

x x x x x x

5392 server-error-version-not-supported

x x x x x x

5393 server-error-device-error

5394 server-error-temporary-error

x x x x x x

5395 server-error-not-accepting-jobs

5396 server-error-busy

x x x x x x

5397 server-error-job-canceled

5398 server-error-multiple-document-jobs-

5399 not-supported

5400

5401

## 5402 14. APPENDIX C: "media" keyword values

5403 Standard keyword values are taken from several sources.

5404 Standard values are defined (taken from DPA[ISO10175] and the Printer MIB[RFC1759]):

- 5405 'default': The default medium for the output device
- 5406 'iso-a4-white': Specifies the ISO A4 white medium
- 5407 'iso-a4-colored': Specifies the ISO A4 colored medium
- 5408 'iso-a4-transparent': Specifies the ISO A4 transparent medium
- 5409 'iso-a3-white': Specifies the ISO A3 white medium
- 5410 'iso-a3-colored': Specifies the ISO A3 colored medium
- 5411 'iso-a5-white': Specifies the ISO A5 white medium
- 5412 'iso-a5-colored': Specifies the ISO A5 colored medium
- 5413 'iso-b4-white': Specifies the ISO B4 white medium
- 5414 'iso-b4-colored': Specifies the ISO B4 colored medium
- 5415 'iso-b5-white': Specifies the ISO B5 white medium
- 5416 'iso-b5-colored': Specifies the ISO B5 colored medium
- 5417 'jis-b4-white': Specifies the JIS B4 white medium
- 5418 'jis-b4-colored': Specifies the JIS B4 colored medium
- 5419 'jis-b5-white': Specifies the JIS B5 white medium
- 5420 'jis-b5-colored': Specifies the JIS B5 colored medium

5421

5422 The following standard values are defined for North American media:

- 5423 'na-letter-white': Specifies the North American letter white medium
- 5424 'na-letter-colored': Specifies the North American letter colored medium
- 5425 'na-letter-transparent': Specifies the North American letter transparent medium
- 5426 'na-legal-white': Specifies the North American legal white medium
- 5427 'na-legal-colored': Specifies the North American legal colored medium

5428

5429 The following standard values are defined for envelopes:

- 5430 'iso-b4-envelope': Specifies the ISO B4 envelope medium
- 5431 'iso-b5-envelope': Specifies the ISO B5 envelope medium
- 5432 'iso-c3-envelope': Specifies the ISO C3 envelope medium
- 5433 'iso-c4-envelope': Specifies the ISO C4 envelope medium
- 5434 'iso-c5-envelope': Specifies the ISO C5 envelope medium
- 5435 'iso-c6-envelope': Specifies the ISO C6 envelope medium
- 5436 'iso-designated-long-envelope': Specifies the ISO Designated Long envelope medium
- 5437 'na-10x13-envelope': Specifies the North American 10x13 envelope medium
- 5438 'na-9x12-envelope': Specifies the North American 9x12 envelope medium



5439       `monarch-envelope`: Specifies the Monarch envelope  
5440       `na-number-10-envelope`: Specifies the North American number 10 business envelope medium  
5441       `na-7x9-envelope`: Specifies the North American 7x9 inch envelope  
5442       `na-9x11-envelope`: Specifies the North American 9x11 inch envelope  
5443       `na-10x14-envelope`: Specifies the North American 10x14 inch envelope  
5444       `na-number-9-envelope`: Specifies the North American number 9 business envelope  
5445       `na-6x9-envelope`: Specifies the North American 6x9 inch envelope  
5446       `na-10x15-envelope`: Specifies the North American 10x15 inch envelope  
5447

5448       The following standard values are defined for the less commonly used media (white-only):

5449       `executive-white`: Specifies the white executive medium  
5450       `folio-white`: Specifies the folio white medium  
5451       `invoice-white`: Specifies the white invoice medium  
5452       `ledger-white`: Specifies the white ledger medium  
5453       `quarto-white`: Specified the white quarto medium  
5454       `iso-a0-white`: Specifies the ISO A0 white medium  
5455       `iso-a1-white`: Specifies the ISO A1 white medium  
5456       `iso-a2-white`: Specifies the ISO A2 white medium  
5457       `iso-a6-white`: Specifies the ISO A6 white medium  
5458       `iso-a7-white`: Specifies the ISO A7 white medium  
5459       `iso-a8-white`: Specifies the ISO A8 white medium  
5460       `iso-a9-white`: Specifies the ISO A9 white medium  
5461       `iso-10-white`: Specifies the ISO A10 white medium  
5462       `iso-b0-white`: Specifies the ISO B0 white medium  
5463       `iso-b1-white`: Specifies the ISO B1 white medium  
5464       `iso-b2-white`: Specifies the ISO B2 white medium  
5465       `iso-b3-white`: Specifies the ISO B3 white medium  
5466       `iso-b6-white`: Specifies the ISO B6 white medium  
5467       `iso-b7-white`: Specifies the ISO B7 white medium  
5468       `iso-b8-white`: Specifies the ISO B8 white medium  
5469       `iso-b9-white`: Specifies the ISO B9 white medium  
5470       `iso-b10-white`: Specifies the ISO B10 white medium  
5471       `jis-b0-white`: Specifies the JIS B0 white medium  
5472       `jis-b1-white`: Specifies the JIS B1 white medium  
5473       `jis-b2-white`: Specifies the JIS B2 white medium  
5474       `jis-b3-white`: Specifies the JIS B3 white medium  
5475       `jis-b6-white`: Specifies the JIS B6 white medium  
5476       `jis-b7-white`: Specifies the JIS B7 white medium  
5477       `jis-b8-white`: Specifies the JIS B8 white medium  
5478       `jis-b9-white`: Specifies the JIS B9 white medium  
5479       `jis-b10-white`: Specifies the JIS B10 white medium  
5480

5481       The following standard values are defined for engineering media (white only):

5482 'a-white': Specifies the engineering A size medium  
5483 'b-white': Specifies the engineering B size medium  
5484 'c-white': Specifies the engineering C size medium  
5485 'd-white': Specifies the engineering D size medium  
5486 'e-white': Specifies the engineering E size medium  
5487

5488 The following standard values are defined for input-trays (from ISO DPA and the Printer MIB):

5489 'top': The top input tray in the printer.  
5490 'middle': The middle input tray in the printer.  
5491 'bottom': The bottom input tray in the printer.  
5492 'envelope': The envelope input tray in the printer.  
5493 'manual': The manual feed input tray in the printer.  
5494 'large-capacity': The large capacity input tray in the printer.  
5495 'main': The main input tray  
5496 'side': The side input tray  
5497

5498 The following standard values are defined for media sizes (from ISO DPA):

5499 'iso-a0': Specifies the ISO A0 size: 841 mm by 1189 mm as defined in ISO 216  
5500 'iso-a1': Specifies the ISO A1 size: 594 mm by 841 mm as defined in ISO 216  
5501 'iso-a2': Specifies the ISO A2 size: 420 mm by 594 mm as defined in ISO 216  
5502 'iso-a3': Specifies the ISO A3 size: 297 mm by 420 mm as defined in ISO 216  
5503 'iso-a4': Specifies the ISO A4 size: 210 mm by 297 mm as defined in ISO 216  
5504 'iso-a5': Specifies the ISO A5 size: 148 mm by 210 mm as defined in ISO 216  
5505 'iso-a6': Specifies the ISO A6 size: 105 mm by 148 mm as defined in ISO 216  
5506 'iso-a7': Specifies the ISO A7 size: 74 mm by 105 mm as defined in ISO 216  
5507 'iso-a8': Specifies the ISO A8 size: 52 mm by 74 mm as defined in ISO 216  
5508 'iso-a9': Specifies the ISO A9 size: 37 mm by 52 mm as defined in ISO 216  
5509 'iso-a10': Specifies the ISO A10 size: 26 mm by 37 mm as defined in ISO 216  
5510 'iso-b0': Specifies the ISO B0 size: 1000 mm by 1414 mm as defined in ISO 216  
5511 'iso-b1': Specifies the ISO B1 size: 707 mm by 1000 mm as defined in ISO 216  
5512 'iso-b2': Specifies the ISO B2 size: 500 mm by 707 mm as defined in ISO 216  
5513 'iso-b3': Specifies the ISO B3 size: 353 mm by 500 mm as defined in ISO 216  
5514 'iso-b4': Specifies the ISO B4 size: 250 mm by 353 mm as defined in ISO 216  
5515 'iso-b5': Specifies the ISO B5 size: 176 mm by 250 mm as defined in ISO 216  
5516 'iso-b6': Specifies the ISO B6 size: 125 mm by 176 mm as defined in ISO 216  
5517 'iso-b7': Specifies the ISO B7 size: 88 mm by 125 mm as defined in ISO 216  
5518 'iso-b8': Specifies the ISO B8 size: 62 mm by 88 mm as defined in ISO 216  
5519 'iso-b9': Specifies the ISO B9 size: 44 mm by 62 mm as defined in ISO 216  
5520 'iso-b10': Specifies the ISO B10 size: 31 mm by 44 mm as defined in ISO 216  
5521 'na-letter': Specifies the North American letter size: 8.5 inches by 11 inches  
5522 'na-legal': Specifies the North American legal size: 8.5 inches by 14 inches  
5523 'executive': Specifies the executive size (7.25 X 10.5 in)  
5524 'folio': Specifies the folio size (8.5 X 13 in)

5525 'invoice': Specifies the invoice size (5.5 X 8.5 in)  
5526 'ledger': Specifies the ledger size (11 X 17 in)  
5527 'quarto': Specifies the quarto size (8.5 X 10.83 in)  
5528 'iso-c3': Specifies the ISO C3 size: 324 mm by 458 mm as defined in ISO 269  
5529 'iso-c4': Specifies the ISO C4 size: 229 mm by 324 mm as defined in ISO 269  
5530 'iso-c5': Specifies the ISO C5 size: 162 mm by 229 mm as defined in ISO 269  
5531 'iso-c6': Specifies the ISO C6 size: 114 mm by 162 mm as defined in ISO 269  
5532 'iso-designated-long': Specifies the ISO Designated Long size: 110 mm by 220 mm as defined in ISO  
5533 269  
5534 'na-10x13-envelope': Specifies the North American 10x13 size: 10 inches by 13 inches  
5535 'na-9x12-envelope': Specifies the North American 9x12 size: 9 inches by 12 inches  
5536 'na-number-10-envelope': Specifies the North American number 10 business envelope size: 4.125  
5537 inches by 9.5 inches  
5538 'na-7x9-envelope': Specifies the North American 7x9 inch envelope size  
5539 'na-9x11-envelope': Specifies the North American 9x11 inch envelope size  
5540 'na-10x14-envelope': Specifies the North American 10x14 inch envelope size  
5541 'na-number-9-envelope': Specifies the North American number 9 business envelope size  
5542 'na-6x9-envelope': Specifies the North American 6x9 envelope size  
5543 'na-10x15-envelope': Specifies the North American 10x15 envelope size  
5544 'monarch-envelope': Specifies the Monarch envelope size (3.87 x 7.5 in)  
5545 'jis-b0': Specifies the JIS B0 size: 1030mm x 1456mm  
5546 'jis-b1': Specifies the JIS B1 size: 728mm x 1030mm  
5547 'jis-b2': Specifies the JIS B2 size: 515mm x 728mm  
5548 'jis-b3': Specifies the JIS B3 size: 364mm x 515mm  
5549 'jis-b4': Specifies the JIS B4 size: 257mm x 364mm  
5550 'jis-b5': Specifies the JIS B5 size: 182mm x 257mm  
5551 'jis-b6': Specifies the JIS B6 size: 128mm x 182mm  
5552 'jis-b7': Specifies the JIS B7 size: 91mm x 128mm  
5553 'jis-b8': Specifies the JIS B8 size: 64mm x 91mm  
5554 'jis-b9': Specifies the JIS B9 size: 45mm x 64mm  
5555 'jis-b10': Specifies the JIS B10 size: 32mm x 45mm

5556 The following standard values are defined for engineering media sizes:

5557 'a': Specifies the engineering A size: 8.5 inches x 11 inches  
5558 'b': Specifies the engineering B size: 11 inches x 17 inches  
5559 'c': Specifies the engineering C size: 17 inches x 22 inches  
5560 'd': Specifies the engineering D size: 22 inches x 34 inches  
5561 'e': Specifies the engineering E size: 34 inches x 44 inches  
5562

## 5563 15. APPENDIX D: Processing IPP Attributes

5564 When submitting a print job to a Printer object, the IPP model allows a client to supply operation and Job  
5565 Template attributes along with the document data. These Job Template attributes in the create request

5566 affect the rendering, production and finishing of the documents in the job. Similar types of instructions  
5567 may also be contained in the document to be printed, that is, embedded within the print data itself. In  
5568 addition, the Printer has a set of attributes that describe what rendering and finishing options which are  
5569 supported by that Printer. This model, which allows for flexibility and power, also introduces the potential  
5570 that at job submission time, these client-supplied attributes may conflict with either:

- 5571 - what the implementation is capable of realizing (i.e., what the Printer supports), as well as
- 5572 - the instructions embedded within the print data itself.

5573

5574 The following sections describe how these two types of conflicts are handled in the IPP model.

## 5575 15.1 Fidelity

5576 If there is a conflict between what the client requests and what a Printer object supports, the client may  
5577 request one of two possible conflict handling mechanisms:

- 5578 1) either reject the job since the job can not be processed exactly as specified, or
- 5579 2) allow the Printer to make any changes necessary to proceed with processing the Job the best it can.

5580

5581 In the first case the client is indicating to the Printer object: "Print the job exactly as specified with no  
5582 exceptions, and if that can't be done, don't even bother printing the job at all." In the second case, the client  
5583 is indicating to the Printer object: "It is more important to make sure the job is printed rather than be  
5584 processed exactly as specified; just make sure the job is printed even if client supplied attributes need to be  
5585 changed or ignored."

5586 The IPP model accounts for this situation by introducing an "ipp-attribute-fidelity" attribute.

5587 In a create request, "ipp-attribute-fidelity" is a boolean operation attribute that is **OPTIONALLY** supplied  
5588 by the client. The value 'true' indicates that total fidelity to client supplied Job Template attributes and  
5589 values is required. The client is requesting that the Job be printed exactly as specified, and if that is not  
5590 possible then the job **MUST** be rejected rather than processed incorrectly. The value 'false' indicates that a  
5591 reasonable attempt to print the Job is acceptable. If a Printer does not support some of the client supplied  
5592 Job Template attributes or values, the Printer **MUST** ignore them or substitute any supported value for  
5593 unsupported values, respectively. The Printer may choose to substitute the default value associated with  
5594 that attribute, or use some other supported value that is similar to the unsupported requested value. For  
5595 example, if a client supplies a "media" value of 'na-letter', the Printer may choose to substitute 'iso-a4' rather  
5596 than a default value of 'envelope'. If the client does not supply the "ipp-attribute-fidelity" attribute, the  
5597 Printer assumes a value of 'false'.

5598 Each Printer implementation **MUST** support both types of "fidelity" printing (that is whether the client  
5599 supplies a value of 'true' or 'false'):

- 5600 - If the client supplies 'false' or does not supply the attribute, the Printer object **MUST** always accept the  
5601 request by ignoring unsupported Job Template attributes and by substituting unsupported values of  
5602 supported Job Template attributes with supported values.

- 5603 - If the client supplies 'true', the Printer object MUST reject the request if the client supplies  
5604 unsupported Job Template attributes.  
5605

5606 Since a client can always query a Printer to find out exactly what is and is not supported, "ipp-attribute-  
5607 fidelity" set to 'false' is useful when:

- 5608 1) The End-User uses a command line interface to request attributes that might not be supported.  
5609 2) In a GUI context, if the End User expects the job might be moved to another printer and prefers a  
5610 sub-optimal result to nothing at all.  
5611 3) The End User just wants something reasonable in lieu of nothing at all.  
5612

## 5613 15.2 Page Description Language (PDL) Override

5614 If there is a conflict between the value of an IPP Job Template attribute and a corresponding instruction in  
5615 the document data, the value of the IPP attribute SHOULD take precedence over the document instruction.  
5616 Consider the case where a previously formatted file of document data is sent to an IPP Printer. In this case,  
5617 if the client supplies any attributes at job submission time, the client desires that those attributes override  
5618 the embedded instructions. Consider the case where a previously formatted document has embedded in it  
5619 commands to load 'iso-a4' media. However, the document is passed to an end user that only has access to a  
5620 printer with 'na-letter' media loaded. That end user most likely wants to submit that document to an IPP  
5621 Printer with the "media" Job Template attribute set to 'na-letter'. The job submission attribute should take  
5622 precedence over the embedded PDL instruction. However, until companies that supply document data  
5623 interpreters allow a way for external IPP attributes to take precedence over embedded job production  
5624 instructions, a Printer might not be able to support the semantics that IPP attributes override the embedded  
5625 instructions.

5626 The IPP model accounts for this situation by introducing a "pdl-override-supported" attribute that describes  
5627 the Printer objects capabilities to override instructions embedded in the PDL data stream. The value of the  
5628 "pdl-override-supported" attribute is configured by means outside the scope of this IPP/1.1 document.

5629 This REQUIRED Printer attribute takes on the following values:

- 5630 - 'attempted': This value indicates that the Printer object attempts to make the IPP attribute values take  
5631 precedence over embedded instructions in the document data, however there is no guarantee.  
5632 - 'not-attempted': This value indicates that the Printer object makes no attempt to make the IPP attribute  
5633 values take precedence over embedded instructions in the document data.  
5634

5635 At job processing time, an implementation that supports the value of 'attempted' might do one of several  
5636 different actions:

- 5637 1) Generate an output device specific command sequence to realize the feature represented by the IPP  
5638 attribute value.  
5639 2) Parse the document data itself and replace the conflicting embedded instruction with a new  
5640 embedded instruction that matches the intent of the IPP attribute value.

- 5641 3) Indicate to the Printer that external supplied attributes take precedence over embedded instructions  
5642 and then pass the external IPP attribute values to the document data interpreter.  
5643 4) Anything else that allows for the semantics that IPP attributes override embedded document data  
5644 instructions.  
5645

5646 Since 'attempted' does not offer any type of guarantee, even though a given Printer object might not do a  
5647 very "good" job of attempting to ensure that IPP attributes take a higher precedence over instructions  
5648 embedded in the document data, it would still be a conforming implementation.

5649 At job processing time, an implementation that supports the value of 'not-attempted' might do one of the  
5650 following actions:

- 5651 1) Simply pre-pend the document data with the PDL instruction that corresponds to the client-supplied  
5652 PDL attribute, such that if the document data also has the same PDL instruction, it will override  
5653 what the Printer object pre-pended. In other words, this implementation is using the same  
5654 implementation semantics for the client-supplied IPP attributes as for the Printer object defaults.  
5655 2) Parse the document data and replace the conflicting embedded instruction with a new embedded  
5656 instruction that approximates, but does not match, the semantic intent of the IPP attribute value.  
5657

5658 Note: The "ipp-attribute-fidelity" attribute applies to the Printer's ability to either accept or reject other  
5659 unsupported Job Template attributes. In other words, if "ipp-attribute-fidelity" is set to 'true', a Job is  
5660 accepted if and only if the client supplied Job Template attributes and values are supported by the Printer.  
5661 Whether these attributes actually affect the processing of the Job when the document data contains  
5662 embedded instructions depends on the ability of the Printer to override the instructions embedded in the  
5663 document data with the semantics of the IPP attributes. If the document data attributes can be overridden  
5664 ("pdl-override-supported" set to 'attempted'), the Printer makes an attempt to use the IPP attributes when  
5665 processing the Job. If the document data attributes can not be overridden ("pdl-override-supported" set to  
5666 'not-attempted'), the Printer makes no attempt to override the embedded document data instructions with the  
5667 IPP attributes when processing the Job, and hence, the IPP attributes may fail to affect the Job processing  
5668 and output when the corresponding instruction is embedded in the document data.

### 5669 15.3 Using Job Template Attributes During Document Processing.

5670 The Printer object uses some of the Job object's Job Template attributes during the processing of the  
5671 document data associated with that job. These include, but are not limited to, "orientation-requested",  
5672 "number-up", "sides", "media", and "copies". The processing of each document in a Job Object MUST  
5673 follow the steps below. These steps are intended only to identify when and how attributes are to be used in  
5674 processing document data and any alternative steps that accomplishes the same effect can be used to  
5675 implement this specification document.

- 5676 1. Using the client supplied "document-format" attribute or some form of document format detection  
5677 algorithm (if the value of "document-format" is not specific enough), determine whether or not the  
5678 document data has already been formatted for printing. If the document data has been formatted,  
5679 then go to step 2. Otherwise, the document data MUST be formatted. The formatting detection  
5680 algorithm is implementation defined and is not specified by this document. The formatting of the

5681 document data uses the "orientation-requested" attribute to determine how the formatted print data  
5682 should be placed on a print-stream page, see section 4.2.10 for the details.

- 5683
- 5684 2. The document data is in the form of a print-stream in a known media type. The "page-ranges"  
5685 attribute is used to select, as specified in section 4.2.7, a sub-sequence of the pages in the print-  
5686 stream that are to be processed and images.
- 5687
- 5688 3. The input to this step is a sequence of print-stream pages. This step is controlled by the "number-up"  
5689 attribute. If the value of "number-up" is N, then during the processing of the print-stream pages,  
5690 each N print-stream pages are positioned, as specified in section 4.2.9, to create a single impression.  
5691 If a given document does not have N more print-stream pages, then the completion of the  
5692 impression is controlled by the "multiple-document-handling" attribute as described in section 4.2.4;  
5693 when the value of this attribute is 'single-document' or 'single-document-new-sheet', the print-stream  
5694 pages of document data from subsequent documents is used to complete the impression.

5695

5696 The size(scaling), position(translation) and rotation of the print-stream pages on the impression is  
5697 implementation defined. Note that during this process the print-stream pages may be rendered to a  
5698 form suitable for placing on the impression; this rendering is controlled by the values of the "printer-  
5699 resolution" and "print-quality" attributes as described in sections 4.2.12 and 4.2.13. In the case N=1,  
5700 the impression is nearly the same as the print-stream page; the differences would only be in the size,  
5701 position and rotation of the print-stream page and/or any decoration, such as a frame to the page,  
5702 that is added by the implementation.

- 5703
- 5704 4. The collection of impressions is placed, in sequence, onto sides of the media sheets. This placement  
5705 is controlled by the "sides" attribute and the orientation of the print-stream page, as described in  
5706 section 4.2.8. The orientation of the print-stream pages affects the orientation of the impression; for  
5707 example, if "number-up" equals 2, then, typically, two portrait print-stream pages become one  
5708 landscape impression. Note that the placement of impressions onto media sheets is also controlled  
5709 by the "multiple-document-handling" attribute as described in section 4.2.4.
- 5710
- 5711 5. The "copies" and "multiple-document-handling" attributes are used to determine how many copies of  
5712 each media instance are created and in what order. See sections 4.2.5 and 4.2.4 for the details.
- 5713
- 5714 6. When the correct number of copies are created, the media instances are finished according to the  
5715 values of the "finishings" attribute as described in 4.2.6. Note that sometimes finishing operations  
5716 may require manual intervention to perform the finishing operations on the copies, especially  
5717 uncollated copies. This document allows any or all of the processing steps to be performed  
5718 automatically or manually at the discretion of the Printer object.

## 5719 16. APPENDIX E: Generic Directory Schema

5720 This section defines a generic schema for an entry in a directory service. A directory service is a means by  
5721 which service users can locate service providers. In IPP environments, this means that IPP Printers can be  
5722 registered (either automatically or with the help of an administrator) as entries of type printer in the

5723 directory using an implementation specific mechanism such as entry attributes, entry type fields, specific  
 5724 branches, etc. IPP clients can search or browse for entries of type printer. Clients use the directory service  
 5725 to find entries based on naming, organizational contexts, or filtered searches on attribute values of entries.  
 5726 For example, a client can find all printers in the "Local Department" context. Authentication and  
 5727 authorization are also often part of a directory service so that an administrator can place limits on end users  
 5728 so that they are only allowed to find entries to which they have certain access rights. IPP itself does not  
 5729 require any specific directory service protocol or provider.

5730 Note: Some directory implementations allow for the notion of "aliasing". That is, one directory entry object  
 5731 can appear as multiple directory entry object with different names for each object. In each case, each alias  
 5732 refers to the same directory entry object which refers to a single IPP Printer object.

5733 The generic schema is a subset of IPP Printer Job Template and Printer Description attributes (sections 4.2  
 5734 and 4.4). These attributes are identified as either RECOMMENDED or OPTIONAL for the directory entry  
 5735 itself. This conformance labeling is NOT the same conformance labeling applied to the attributes of IPP  
 5736 Printers objects. The conformance labeling in this Appendix is intended to apply to directory templates and  
 5737 to IPP Printer implementations that subscribe by adding one or more entries to a directory.  
 5738 RECOMMENDED attributes SHOULD be associated with each directory entry. OPTIONAL attributes  
 5739 MAY be associated with the directory entry (if known or supported). In addition, all directory entry  
 5740 attributes SHOULD reflect the current attribute values for the corresponding Printer object.

5741 The names of attributes in directory schema and entries SHOULD be the same as the IPP Printer attribute  
 5742 names as shown.

5743 In order to bridge between the directory service and the IPP Printer object, one of the RECOMMENDED  
 5744 directory entry attributes is the Printer object's "printer-uri-supported" attribute. The IPP client queries the  
 5745 "printer-uri-supported" attribute in the directory entry and then addresses the IPP Printer object using one of  
 5746 its URIs. The "uri-security-supported" attribute identifies the protocol (if any) used to secure a channel.

5747 The following attributes define the generic schema for directory entries of type PRINTER:

5748	printer-uri-supported	RECOMMENDED	Section 4.4.1
5749	uri-authentication-supported	RECOMMENDED	Section 4.4.2
5750	uri-security-supported	RECOMMENDED	Section 4.4.3
5751	printer-name	RECOMMENDED	Section 4.4.4
5752	printer-location	RECOMMENDED	Section 4.4.5
5753	printer-info	OPTIONAL	Section 4.4.6
5754	printer-more-info	OPTIONAL	Section 4.4.7
5755	printer-make-and-model	RECOMMENDED	Section 4.4.9
5756	ipp-versions-supported	RECOMMENDED	Section 4.4.14
5757	multiple-document-jobs-supported	OPTIONAL	Section 4.4.16
5758	charset-supported	OPTIONAL	Section 4.4.18
5759	generated-natural-language-		
5760	supported	OPTIONAL	Section 4.4.20
5761	document-format-supported	RECOMMENDED	Section 4.4.22
5762	color-supported	RECOMMENDED	Section 4.4.26
5763	compression-supported	RECOMMENDED	Section 4.4.32



5764	pages-per-minute	OPTIONAL	Section 4.4.36
5765	pages-per-minute-color	OPTIONAL	Section 4.4.37
5766			
5767	finishings-supported	OPTIONAL	Section 4.2.6
5768	number-up-supported	OPTIONAL	Section 4.2.7
5769	sides-supported	RECOMMENDED	Section 4.2.8
5770	media-supported	RECOMMENDED	Section 4.2.11
5771	printer-resolution-supported	OPTIONAL	Section 4.2.12
5772	print-quality-supported	OPTIONAL	Section 4.2.13

5773

## 5774 17. APPENDIX F: Differences between the IPP/1.0 and IPP/1.1 "Model and Semantics" Documents

5775 This Appendix is divided into two lists that summarize the differences between IPP/1.1 (this document) and  
5776 IPP/1.0 [RFC2566]. The section numbers refer to the numbers in this document which in some cases have  
5777 changed from RFC 2566. When a change affects multiple sections, the item is listed once in the order of  
5778 the first section affected and the remaining affected section numbers are indicated.

5779 The first list contains extensions and clarifications and the second list contains changes in semantics or  
5780 conformance. However, client and IPP object implementations of IPP/1.0 MAY implement any of the  
5781 extensions and clarifications in this document.

5782 The following extensions and clarifications have been incorporated into this document:

- 5783 1. Section 2.1 - clarified that the term "client" can be either contained in software controlled by an end  
5784 user or a part of a print server that controls devices.
- 5785 2. Section 2 - clarified that the term "IPP object" and "Printer object" can either be embedded in a  
5786 device object or part of a print server that accepts IPP requests.
- 5787 3. Section 2.4 - added the description of the new "uri-authentication-supported" Printer Description  
5788 attribute.
- 5789 4. Section 3.1.3, 3.1.6, 3.2.5.2, and 3.2.6.2 - clarified the error handling for operation attributes that  
5790 have their own status code.
- 5791 5. Section 3.1.6 - reorganized this section into sub-sections to separately describe "status-code",  
5792 "status-message", "detailed-status-message", and "document-access-error" attributes.
- 5793 6. Section 3.1.6.1 - clarified the error status codes and their relationship to operation attributes.
- 5794 7. Section 3.1.6.3 - Added the OPTIONAL "detailed-status-message (text(MAX))" operation attribute  
5795 to provide additional more detailed information about a response.
- 5796 8. Section 3.1.6.4 and 3.2.2 - Added the OPTIONAL "document-access-error (text(MAX))" operation  
5797 attribute for use with Print-URI and Send-URI responses.
- 5798 9. Sections 3.1.7 - Added this new section to clarify returning Unsupported Attributes for all  
5799 operations, including only returning attributes that were in the request. Moved the text from section  
5800 3.2.1.2 Unsupported Attributes to this section.
- 5801 10. Sections 3.1.7 and 4.1 - clarified the encoding of the "out-of-band" 'unsupported' and 'unknown'  
5802 values.
- 5803 11. Section 3.1.8 - clarified that only the version number parameter will be carried forward into future  
5804 major or minor versions of the protocol.
- 5805 12. Section 3.1.8 - relaxed the requirements to increment the major version number in future versions of  
5806 the Model and Semantics document.
- 5807 13. Section 3.1.9, and 3.2.5 - added the 'processing' state to the list of job states that a job can be in after  
5808 a Create-Job operation.
- 5809 14. Section 3.1.9 - clarified that a non-spooling Printer MAY accept zero or more subsequent jobs while  
5810 processing a job and flow control them down. Subsequent create requests are rejected with the  
5811 'server-error-busy' error status.

- 5812 15. Section 3.2.1.1 - clarified the validation of the "compression" operation attribute and its relationship  
5813 to the validation of the "document-format" attribute and returning Unsupported Attributes.
- 5814 16. Sections 3.2.1.1, 4.3.8, 13.1.4.16, and 13.1.4.17 - added the 'client-error-compression-not-  
5815 supported', 'client-error-compression-error' status codes and the 'unsupported-compression' and  
5816 'compression-error' job-state-reasons.
- 5817 17. Sections 3.2.1.1 and 4.3.8 - added 'unsupported-document-format' and 'document-format-error' job-  
5818 state-reasons.
- 5819 18. Sections 3.2.2, 4.3.8 and 13.1.4.19 - added 'client-error-document-access-error' status code and  
5820 'document-access-error' job state reason.
- 5821 19. Section 3.2.5.2 and 3.2.6.2 - clarified that the Unsupported Attributes group MUST NOT include  
5822 attributes not requested in the Get-Printer-Attributes request.
- 5823 20. Section 3.2.6 - clarified that "limit" takes precedence over "which-jobs" and "my-jobs".
- 5824 21. Section 3.2.6.2 - clarified that Get-Jobs returns 'successful-ok' when no jobs to return.
- 5825 22. Sections 3.2.7, 3.2.8, and 3.2.9 - added the OPTIONAL Pause-Printer, Resume-Printer, and Purge-  
5826 Jobs operations
- 5827 23. Section 3.3.1 - clarified that the authorization required for a Send-Document request MUST be the  
5828 same user as the Create-Job or an operator.
- 5829 24. Sections 3.3.5, 3.3.6, and 3.3.7 - added the OPTIONAL Hold-Job, Release-Job, and Restart-Job  
5830 operations.
- 5831 25. Section 4.1 - clarified that the encoding of the out-of-band values are specified in the Encoding and  
5832 Transport" document.
- 5833 26. Section 4.1.9.1 - clarified that 'application/octet-stream' auto-sensing can happen at create request  
5834 time and/or job/document processing time.
- 5835 27. Section 4.1.14 - clarified that the localization of dateTime by the client includes the time zone.
- 5836 28. Section 4.2 - clarified that xxx-supported have multiple keywords and/or names by adding  
5837 parentheses to the table to give: (1setOf (type3 keyword | name))
- 5838 29. Section 4.2.2 - added the 'indefinite' keyword value to the "job-hold-until" attribute for use with the  
5839 create operations and Hold-Job and Restart-Job operations.
- 5840 30. Section 4.2.6 - added more enum values to the "finishings" Job Template attribute.
- 5841 31. Section 4.3.7 - added that a forwarding server that cannot get any job state MAY return the job's  
5842 state as 'completed', provided that it also return the new 'queued-in-device' job state reason.
- 5843 32. Section 4.3.7.2 - added the Partitioning of Job States section to clarify the concepts of Job  
5844 Retention, Job History, and Job Removal.
- 5845 33. Section 4.3.8 - added 'job-data-insufficient' job state reason to indicate whether sufficient data has  
5846 arrived for the document to start to be processed.
- 5847 34. Section 4.3.8 - added 'document-access-error' job state reason to indicate an access error of any kind.
- 5848 35. Section 4.3.8 - added 'job-queued-for-marker' job state reason to indicate whether the job has  
5849 completed some processing and is waiting for the marker.
- 5850 36. Section 4.3.8 - added 'unsupported-compression' and 'compression-error' job state reasons to  
5851 indicate compression not supported or compression processing error after the create has been  
5852 accepted.
- 5853 37. Section 4.3.8 - added 'unsupported-document-format' and 'document-format-error' job state reasons  
5854 to indicate document not supported or document format processing error after the create has been  
5855 accepted.

- 5856 38. Section 4.3.8 - added 'queued-in-device' job state reason to indicate that a job as been forwarded to a  
5857 print system or device that does not provide any job status.
- 5858 39. Section 4.3.10 - added "job-detailed-status-messages (1setOf text(MAX)) for returning detailed  
5859 error messages.
- 5860 40. Section 4.3.11 - added the "job-document-access-errors (1setOf text(MAX))
- 5861 41. Section 4.3.14.2 - clarified that the time recorded is the first time processing since the create  
5862 operation or the Restart-Job operation.
- 5863 42. Section 4.3.14.2 and 4.3.14.3 - clarified that the out-of-band value 'no-value' is returned if the job  
5864 has not started processing or has not completed, respectively.
- 5865 43. Section 4.3.14 - Added the OPTIONAL "date-time-at-creation", "date-time-at-processing", and  
5866 "date-time-at-completed" Event Time Job Description attributes
- 5867 44. Section 4.4.3 - added the 'tls' value to "uri-security-supported" attribute.
- 5868 45. Section 4.4.3 - clarified "uri-security-supported" is orthogonal to Client Authentication so that 'none'  
5869 does not exclude Client Authentication.
- 5870 46. Section 4.4.11 - simplified the "printer-state" descriptions while generalizing to allow high end  
5871 devices that interpret one or more jobs while marking another. Indicated that 'spool-area-full' and  
5872 'stopped-partly' "printer-state-reasons" may be used to provide further state information.
- 5873 47. Section 4.4.12 - added the 'moving-to-paused' keyword value to the "printer-state-reasons" attribute  
5874 for use with the Pause-Job operation.
- 5875 48. Section 4.4.12 - replaced the duplicate 'marker-supply-low' keyword with the missing 'toner-empty'  
5876 keyword for the "printer-state-reasons" attribute. (This correction was also made before RFC 2566  
5877 was published).
- 5878 49. Section 4.4.12 - clarified 'spool-area-full' "printer-state-reasons" to include non-spooling printers to  
5879 indicate when it can and cannot accept another job.
- 5880 50. Section 4.4.15 - added the enum values to the "operations-supported" attribute for the new  
5881 operations. Clarified that the values of this attribute are encoded as any enum, namely 32-bit values.
- 5882 51. Section 4.4.30 - clarified that the dateTime value of "printer-current-time" is on a "best efforts  
5883 basis". If a proper date-time cannot be obtained, the implementation returns the 'no-value' out-of-  
5884 band value. Also clarified that the time zone NEED NOT be the time zone that the people near the  
5885 device use and that the client SHOULD display the dateTime attributes in the user's local time.
- 5886 52. Sections 4.4.36 and 4.4.37 - added the OPTIONAL "pages-per-minute" and "pages-per-minute-  
5887 color" Printer Description attributes.
- 5888 53. Section 5.1 - clarified that the client conformance requirements apply to clients controlled by an end  
5889 user and clients in servers.
- 5890 54. Section 5.1 - clarified that any response MAY contain additional attribute groups, attributes,  
5891 attribute syntaxes, or attribute values.
- 5892 55. Section 5.1 - clarified that a client SHOULD do its best to prevent a channel from being closed by a  
5893 lower layer when the channel is flow controlled off by the IPP Printer.
- 5894 56. Section 5.2 - clarified that the IPP object requirements apply to objects embedded in devices or that  
5895 are parts of servers.
- 5896 57. Section 5.2.2 - clarified that IPP objects MAY return operation responses that contain attribute  
5897 groups, attribute names, attribute syntaxes, attribute values, and status codes that are extensions to  
5898 this standard.
- 5899 58. Section 8.3 - clarified the use of URIs for each Client Authentication mechanism.
- 5900 59. Section 8.5 - added the security discussion around the new operator/administrator operations.

- 5901 60. Section 13.1.4.16 - added client-error-compression-not-supported (0x040F)  
5902 61. Section 13.1.4.17 - added client-error-compression-error (0x0410)  
5903 62. Section 13.1.4.18 - added client-error-document-format-error (0x0411)  
5904 63. Section 13.1.4.19 - added client-error-document-access-error (0x0412)  
5905 64. Section 13.1.5.10 - added server-error-multiple-document-jobs-not-supported (0x0509)  
5906 65. Section 14 - added 'a-white', 'b-white', 'c-white', 'd-white', and 'e-white' and clarified that the existing  
5907 'a', 'b', 'c', 'd', and 'e' values are size values.  
5908 66. Section 16 - added the OPTIONAL "pages-per-minute" and "pages-per-minute-color" Printer  
5909 attributes to the Directory schema.  
5910 67. Section 16 - added OPTIONAL "multiple-document-jobs-supported" to the Directory schema.  
5911 68. Section 16 - added RECOMMENDED "uri-authentication-supported", "ipp-versions-supported",  
5912 and "compression-supported" to the Directory schema.

5913 The following changes in semantics and/or conformance have been incorporated into this document:

- 5914 1. Section 3.1.8, 5.2.4, and 13.1.5.4 - Clients and IPP objects MUST support version 1.1  
5915 conformance requirements and SHOULD support version 1.0 conformance requirements. Also  
5916 clarified that IPP Printers MUST accept '1.1' requests and SHOULD accept '1.x' requests.  
5917 2. Section 3.2.1.1 and section 4.4.32 - changed the "compression" operation and the "compression-  
5918 supported" Printer Description attribute from OPTIONAL to REQUIRED.  
5919 3. Sections 3.2.1.2 and 4.3.8 - changed "job-state-reasons" from RECOMMENDED to REQUIRED,  
5920 so that "job-state-reasons" MUST be returned in create operation responses.  
5921 4. Sections 3.2.4, 3.3.1, 4.4.16, and 16 - changed Create-Job/Send-Document so that they MAY be  
5922 implemented while only supporting one document jobs. Added the "multiple-document-jobs-  
5923 supported" boolean Printer Description attribute to indicate whether Create-Job/Send-  
5924 Document support multiple document jobs or not. Added to the Directory schema.  
5925 5. Section 4.1.9 - deleted 'text/plain; charset=iso-10646-ucs-2', since binary is not legal with the 'text'  
5926 type.  
5927 6. Section 4.2.4 - indicated that the "multiple-document-handling" Job Template attribute MUST be  
5928 supported with at least one value if the Printer supports multiple documents per job  
5929 7. Section 4.3.7.2 - indicated that the 'job-restartable' job state reason SHOULD be supported if the  
5930 Restart-Job operation is supported.  
5931 8. Section 4.3.8 - changed "job-state-reasons" from RECOMMENDED to REQUIRED.  
5932 9. Section 4.3.8 - clarified the conformance of the values of the "job-state-reasons" attribute by  
5933 copying conformance requirements from other sections of the document so that it is clear from  
5934 reading the definition of "job-state-reasons" which values MUST or SHOULD be supported.  
5935 The 'none', 'unsupported-compression', and 'unsupported-document-format' values MUST be  
5936 supported. The 'job-hold-until-specified' SHOULD be specified if the "job-hold-until" Job  
5937 Template is supported. The following values SHOULD be supported: 'job-canceled-by-user',  
5938 'aborted-by-system', and 'job-completed-successfully'. The 'job-canceled-by-operator' SHOULD  
5939 be supported if the implementation permits canceling by other than the job owner. The 'job-  
5940 canceled-at-device' SHOULD be supported if the device supports canceling jobs at the console.  
5941 The 'job-completed-with-warnings' SHOULD be supported, if the implementation detects  
5942 warnings. The 'job-completed-with-errors' SHOULD be supported if the implementation  
5943 detects errors. The 'job-restartable' SHOULD be supported if the Restart-Job operation is  
5944 supported.

- 5945 10. Section 4.3.14 - changed the "time-at-creation", "time-at-processing", and "time-at-completed"  
5946 Event Time Job Description attributes from OPTIONAL to REQUIRED.
- 5947 11. Section 4.3.14.4 - added the REQUIRED "job-printer-up-time (integer(1:MAX))" Job Description  
5948 attribute as an alias for "printer-up-time" to reduce number of operations to get job times.
- 5949 12. Section 4.4.2 - added the REQUIRED "uri-authentication-supported (1setOf type2 keyword)"  
5950 Printer Description attribute to describe the Client Authentication used by each Printer URI.
- 5951 13. Section 4.4.12 - changed "printer-state-reasons" Printer Description attribute from OPTIONAL to  
5952 REQUIRED.
- 5953 14. Section 4.4.12 - changed 'paused' value of "printer-state-reasons" to MUST if Pause-Printer  
5954 operation is supported.
- 5955 15. Section 4.4.14 - added the REQUIRED "ipp-versions-supported (1setOf keyword)" Printer  
5956 Description attribute, since IPP/1.1 Printers do not have to support version '1.0' conformance  
5957 requirements. Section 4.4.16 - added the "multiple-document-jobs-supported (boolean)" Printer  
5958 Description attribute so that a client can tell whether a Printer that supports Create-Job/Send-  
5959 Document supports multiple document jobs or not. This attribute is REQUIRED if the Create-  
5960 Job operation is supported.
- 5961 16. Section 4.4.24 - changed the "queued-job-count" Printer Description attribute from  
5962 RECOMMENDED to REQUIRED.
- 5963 17. Section 4.4.32 - changed "compression-supported (1setOf type3 keyword)" Printer Description  
5964 attribute from OPTIONAL to REQUIRED.
- 5965 18. Section 5.1 - changed the client security requirements from RECOMMENDED non-standards  
5966 track SSL3 to MUST support Client Authentication as defined in the IPP/1.1 Encoding and  
5967 Transport document [IPP-PRO]. A client SHOULD support Operation Privacy and Server  
5968 Authentication as defined in the IPP/1.1 Encoding and Transport document [IPP-PRO].
- 5969 19. Section 5.2.7 - changed the IPP object security requirements from OPTIONAL non-standards track  
5970 SSL3 to SHOULD contain support for Client Authentication as defined in the IPP/1.1 Encoding  
5971 and Transport document [IPP-PRO]. A Printer implementation MAY allow an administrator to  
5972 configure the Printer so that all, some, or none of the users are authenticated. An IPP Printer  
5973 implementation SHOULD contain support for Operation Privacy and Server Authentication as  
5974 defined in the IPP/1.1 Encoding and Transport document [IPP-PRO]. A Printer implementation  
5975 MAY allow an administrator to configure the degree of support for Operation Privacy and  
5976 Server Authentication. Security MUST NOT be compromised when the client supplies a lower  
5977 version-number in a request.

5978 See also the "IPP/1.1 Encoding and Transport" [IPP-PRO] document for differences between IPP/1.0  
5979 [RFC2565] and IPP/1.1 [IPP-PRO].

## 5980 18. Full Copyright Statement

5981 Copyright (C) The Internet Society (1999). All Rights Reserved.

5982 This document and translations of it may be copied and furnished to others, and derivative works that  
5983 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and  
5984 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and  
5985 this paragraph are included on all such copies and derivative works. However, this document itself may not

5986 be modified in any way, such as by removing the copyright notice or references to the Internet Society or  
5987 other Internet organizations, except as needed for the purpose of developing Internet standards in which  
5988 case the procedures for copyrights defined in the Internet Standards process must be followed, or as  
5989 required to translate it into languages other than English.

5990 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its  
5991 successors or assigns.

5992 This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET  
5993 SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES,  
5994 EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE  
5995 OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED  
5996 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

5997